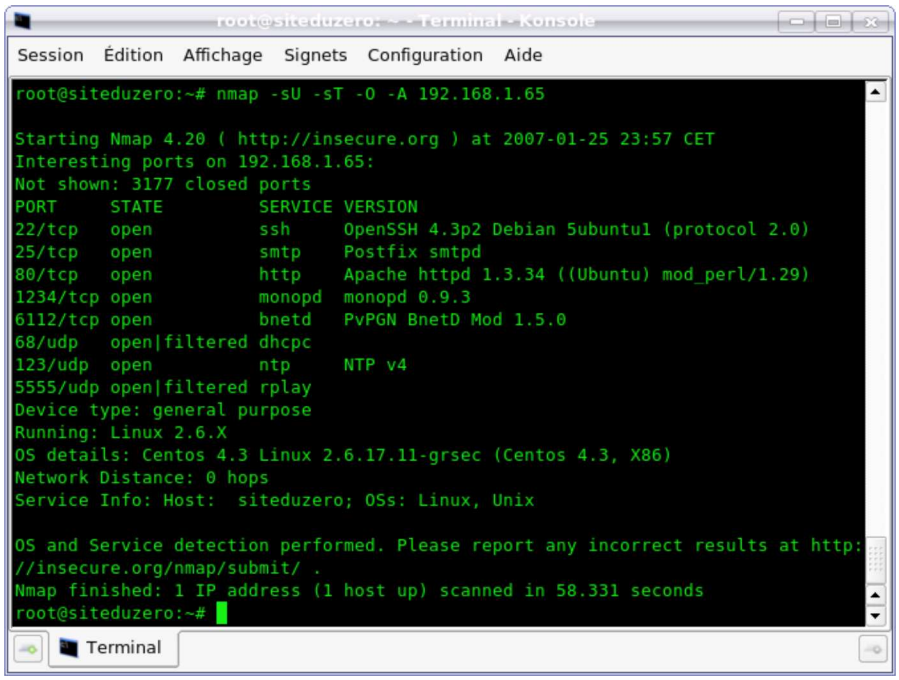
Un livre de Wikilivres.



Sécurité des systèmes informatiques

Une version à jour et éditable de ce livre est disponible sur Wikilivres,
une bibliothèque de livres pédagogiques, à l'URL :
http://fr.wikibooks.org/wiki/S%C3%A9curit%C3%A9_des_syst%C3%A8mes_informatiques

---

Sécurité informatique/Introduction

La sécurité-confidentialité

Nous présentons dans cette section la terminologie relative à la sécurité. La sécurité peut être vue comme une propriété particulière de la *sûreté de fonctionnement*. La sécurité correspond alors aux attributs suivants pour un système :

- la *confidentialité*, c'est à dire la non-occurrence de divulgation non-autorisées de l'information ;
- l'*intégrité*, c'est à dire la non-occurrence d'altérations inappropriées de l'information ;
- et la *disponibilité*, qui correspond au fait d'être prêt à l'utilisation.

L'association de la confidentialité, de l'intégrité et de la disponibilité correspond à la sécurité telle que nous l'abordons dans ce document. On peut la désigner sous le nom de sécurité-confidentialité en fonction de son attribut le plus distinctif pour la distinguer d'une autre propriété de la sûreté de fonctionnement, la sécurité-innocuité. (Cette dernière correspond à la non-occurrence dans le système de conséquences catastrophiques pour l'environnement.) Ce document traitant exclusivement de la sécurité-confidentialité, nous utiliserons la désignation directe sécurité pour la nommer.

Dans le cadre de la sûreté de fonctionnement des systèmes informatiques ou des systèmes d'information, les moyens utilisables pour traiter de la sécurité-confidentialité, notamment face à des malveillances, peuvent être organisés autour des points suivants :

- *Prévention* : la prévention des fautes vise à empêcher l'occurrence ou l'introduction de fautes.
- *Tolérance* : la tolérance aux fautes correspond à un ensemble de moyens destinés à assurer qu'un système remplit sa fonction en dépit des fautes.
- *Élimination* : l'élimination des fautes vise à réduire le nombre ou la sévérité des fautes.
- *Prévision* : la prévision des fautes vise l'estimation de la présence, la création et les conséquences des fautes.

Voies d'action

Plusieurs voies sont disponibles pour aborder les problèmes de sécurité des systèmes d'information ; nous en avons identifiées dans la liste ci-dessous, organisée de manière empirique, notamment par rapport aux différents points abordés dans l'ensemble de ce texte :

- Actions non-techniques
  - Gestion de la SSI
  - Délégation et habilitation des personnes
  - Contrats
  - Formation / Sensibilisation
- Actions de protection
  - Réseau
  - Système
  - Applications
- Actions de surveillance
  - Détection d'intrusion
  - Observation
- Agressions
  - Attaques
  - Audit / Enquête
  - Tests d'intrusion

Technologies concrètes les plus courantes

Dans la pratique, certaines technologies ou certaines pratiques, parfois très ponctuelles dominent largement le quotidien de la sécurité des systèmes informatiques actuels :

- Firewall (ou pare-feu) : équipements de filtrage réseau au niveau TCP/IP.
- Relayage et filtrage HTTP : solutions (généralement logicielles) de contrôle et de filtrage des flux réseau des applications appuyées sur des relais (ou proxy) dont les plus courantes concernent le protocole HTTP.
- Détection d'intrusion : sondes repérant, généralement à partir du trafic réseau, des attaques au moment où elles surviennent.
- Systèmes d'authentification : moyens matériels ou logiciels de vérification de l'identité des utilisateurs (humains ou machines).
- VPN : solution de chiffrement de flux réseau (souvent avec encapsulation des flux).
- Protection des applications : verrous et protections mis en œuvre directement par les applications elles-mêmes.
- Antivirus (poste de travail et flux) : systèmes de détection des virus (codes malveillants dont la caractéristique principale est l'auto-propagation) au niveau des postes de travail ou des flux (messagerie, HTTP, etc.).
- Pratiques d'administration : organisation des procédures d'administration avec (ou sans) prise en compte de la sécurité.
- Observation et surveillance réseau : analyse ponctuelle des flux réseau et surveillance générale des systèmes (en lien avec les pratiques d'exploitation orientées vers la fiabilité).

Parmi les différentes techniques disponibles, certaines sont très largement répandues, parfois à l'exclusion d'autres éventuellement disponibles dans l'état de l'art. On recense alors notamment :

- les techniques d'authentification par nom d'utilisateur et mot de passe (à tous les niveaux : systèmes d'exploitation, applications, accès réseau, etc.) ;
- les systèmes de détection d'attaques réseau à base de signatures ;
- les « cartes à puces » (cartes ou clefs) pour l'authentification « forte » (dans le domaine bancaire, administratif, ou la monétique) ;
- les algorithmes de cryptographie : RSA, DSA, 3DES, AES.

Certaines techniques semblent émergentes à l'heure actuelle, notamment :

- la biométrie (du doigt).

Commentaires

Une vulgarisation rapide conduirait certainement à présenter presque exclusivement les sujets suivants pour le domaine de la sécurité informatique : l'antivirus, le firewall, le filtrage d'URL et peut-être la propagation des correctifs (patch). Il s'agirait donc presque exclusivement de techniques de prévention, parfois déployées de manière un peu aveugle.

Ces déploiements se complètent aussi en général d'une prise en compte de la sécurité du système informatique au niveau de l'organisation de l'entreprise, ne serait-ce que par la création d'une fonction de type RSSI. C'est le signe d'une meilleure maturité dans la compréhension de la problématique (qui ne se limite malheureusement pas à l'application de recettes et l'installation d'équipements tout prêts) ; mais cela ne permet généralement pas à l'heure actuelle d'aborder complètement toutes les facettes de la sécurité informatique sur le système réel (de l'authentification à la détection d'intrusion sur le réseau local par exemple).

En effet, outre les contraintes de coûts (à la fois en matériels, logiciels et personnels pour les administrer), les difficultés d'organisation (par exemple pour la formation ou la garantie de l'indépendance des administrateurs sécurité), ainsi que la complexité technique d'un diagnostic valide (pouvant combiner des notions de cryptographie, de réseau, de système d'exploitation, et, in fine, le besoin de le défendre devant un tribunal) ; il reste l'obstacle fondamental lié au fait que la sécurité n'est généralement pas la finalité première d'un système informatique civil. Dans la plupart des cas, les exigences de sécurité sont perçues comme secondaires par rapport aux besoins initiaux, et cette attitude, qu'elle soit justifiée ou non, mène parfois à une réutilisation ou des réactions un peu trop automatiques. [1]

Malgré les présentations plus ou moins théorisantes, la réalité de la sécurité informatique à l'heure actuelle reste principalement une juxtaposition de deux choses : la mise en place de recettes industrielles toutes prêtes à l'efficacité souvent difficile à évaluer, et les efforts pratiques de personnalités individuelles aux compétences parfois très rigoureuses mais largement basées sur une connaissance technique autodidacte. Les efforts généraux de l'industrie qui se développe autour de la thématique sécurité visent bien évidemment quand même à améliorer au maximum la qualité technique des recettes proposées et à les faire évoluer vers une organisation et des méthodes plus générales et plus efficaces (mais dont l'évaluation reste toutefois pour l'instant un sujet assez mystérieux). Pourtant, les efforts concrets des « sysadmin/guru/hackers » ne sont pas à négliger car, dans la pratique, ils justifient pour une grande part la confiance que l'on peut accorder - ou, en leur absence, qu'il n'est pas judicieux d'accorder - à la sécurité des systèmes informatiques actuels. Bien qu'ignorés par les méthodologies industrielles, il est clair qu'au coeur d'un système informatique de confiance nous trouverons encore toujours à l'heure actuelle avant tout des individus qui contribuent à la sécurité du système (sans nécessairement en avoir officiellement la responsabilité). Ils s'inscrivent dans une logique de prise en compte ouverte des problèmes et de leurs solutions certes parfois désorganisée au niveau global, mais génératrice d'efforts pratiques réellement orientés vers l'amélioration de la sécurité ; en puisant peut-être une motivation aux mêmes sources que ceux qui, parfois, mettent en danger ces mêmes systèmes. La sécurité informatique est encore une affaire de passionnés. Que ce soit heureux ou malheureux, pour l'instant, nous ne sommes pas vraiment en mesure d'en juger.

Notes

1. Par exemple, même si parfois, les besoins fonctionnels eux-mêmes sont mis en danger par l'insécurité du système et si les risques sont inacceptables, rares sont les situations où il est possible de provoquer une prise de conscience suffisamment aigue d'un risque pour prononcer l'arrêt d'un système. C'est d'ailleurs une constatation qui militerait fortement en faveur du développement de techniques utilisant une approche basée sur la tolérance (aux intrusions).

Sécurité informatique/Le domaine SSI

Fonctions du RSSI

Les fiches de définition de poste du Cigref identifient les grandes missions suivantes pour la fonction de « Responsable de la Sécurité du Système d'Information », fréquemment abrégée « RSSI » : Définition de la politique de sécurité : construire le référentiel normatif de l'organisation vis à vis de la sécurité informatique, en accord avec les objectifs de la direction générale et les contraintes de mise en place ou les risques identifiés.

- Analyse de risques : identifier et évaluer les risques liés au système d'information (et notamment son informatisation).
- Sensibilisation et formation aux enjeux de la sécurité : accompagner les utilisateurs et les informaticiens de l'organisation pour mettre en lumière les enjeux liés à la sécurité et les moyens d'y répondre.
- Étude des moyens et préconisations : être une force de proposition de moyens techniques permettant d'atteindre les objectifs de sécurité de l'organisation ou de pallier aux risques inacceptables, notamment par le biais d'études techniques.
- Audit et contrôle : contrôler la mise en place des règles de sécurité, vérifier le niveau de vulnérabilité réel du système d'information, et éventuellement effectuer (du point de vue technique) des enquêtes ou des audits internes si besoin.
- Veille technologique et prospective : effectuer un suivi général des offres du marché de la sécurité, mais aussi des évolutions théoriques de ce secteur, et assurer un suivi des vulnérabilités et des alertes de sécurité concernant les systèmes informatiques auprès des entités agissant sur ce thème (constructeurs, CERT, etc.).

Ces missions conduisent donc à donner au RSSI des rôles de conseil, d'assistance, d'information, de formation et d'alerte. Il s'agit de rôles demandant à la fois des capacités d'intervention variées et des compétences multi-disciplinaires ; ce qui rend la fonction assez difficile à remplir dans son ensemble. Dans la mesure du possible, ces missions doivent être accomplies dans une structure indépendante de la direction informatique. [1]

Organisation

Les principaux composants de l'organisation d'une entreprise pour la gestion de la SSI sont les suivants :

- Un « responsable » (RSSI) : qui assure seul la coordination sur ce thème, ou qui gère éventuellement une équipe technique chargée des systèmes de sécurité informatiques dédiés à ce domaine.
- Comité de sécurité informatique : un comité regroupant les acteurs décisionnaires sur le domaine de la SSI (direction générale, direction informatique, RSSI notamment) et faisant autorité pour les questions de politique générale et de moyens matériels affectés à la SSI.
- Groupes de travail : des groupes de travail opérationnels sont généralement nécessaires, notamment par thèmes, pour faire progresser les différents sujets impliqués dans l'atteinte des objectifs de sécurité (réseau, poste de travail, systèmes, etc.)
- Veille technologique : la veille technologique des alertes de sécurité (vulnérabilités, menaces, etc.) demande généralement une structure identifiée qui peut être directement réalisée par une équipe technique SSI (ou le RSSI) mais qui peut également bénéficier du filtre de documentalistes professionnelles ou de prestataires extérieurs.
- Suivi de la sécurité opérationnelle : la gestion quotidienne de la sécurité peut impliquer un travail d'exploitation et de suivi (notamment des équipements de sécurité).
- Surveillance et contrôle : la surveillance continue du système informatique, sous l'angle par exemple de la détection d'intrusion ou du contrôle de la conformité des systèmes aux règles de sécurité définies sont une autre facette du travail technique quotidien consacré à la sécurité du système informatique.
- Sensibilisation des utilisateurs : la sensibilisation des utilisateurs aux problèmes et aux efforts de sécurité est une action importante dans la pratique, à mener en général avec le service ou les actions de communication interne.
- Autorisation et gestion des habilitations : la délivrance des autorisations aux différents employés et la gestion (éventuellement manuelle) des habilitations associées peut constituer une activité déterminante dans le domaine de la sécurité, cette fois-ci au sens large en incluant les droits d'accès et les fonctions des personnes dans l'organisation (souvent en coordination avec la gestion des ressources humaines).
- projet *X* : pour les différents projets, des équipes spécifiquement chargées de la prise en compte des exigences de sécurité, ou des réalisations techniques associées peuvent être identifiées ; en règle générale, seul les projets d'envergure, les projets risqués, ou les projets spécifiques à la sécurité nécessitent (ou font l'effort de constituer) une équipe spécifique sur ce thème.
- Gestion de crise : une cellule de crise peut éventuellement être constituée en prévision de réaction à des situations exceptionnelles du point de vue de la sécurité informatique, suivant le niveau de risque et les enjeux associés au système d'information de l'organisation.

Documents SSI

On peut identifier un certains nombre de documents entourant la gestion de la sécurité :

- Politique de sécurité (PSSI) : C'est le document de plus haut niveau fixant notamment les objectifs de sécurité détaillés de l'entreprise (et donc les décisions politiques de protection de ses actifs par rapport aux risques identifiés ou éventuels) et les règles de sécurité à mettre en place pour les atteindre. Validé par la direction générale, ce document permet d'organiser et de légitimer la mise en place de l'organisation relative à la SSI et des recommandations plus spécifiques qui découlent de la politique de sécurité.
- Spécifications de sécurité : Dans un certain nombre de domaines, il est en effet utile de décliner les règles de haut niveau adoptées dans la PSSI pour les adapter à un contexte particulier. Par exemple, dans le domaine contractuel, la PSSI peut se trouver décliner dans un modèle de clause de sécurité pour les marchés établis par l'entreprise avec ses sous-traitants (notamment s'il s'agit de marchés publics) rédigé en concertation avec le service juridique de l'entreprise. Dans le domaine de la surveillance interne, la PSSI peut devoir être précisée pour clarifier les modalités de surveillance des salariés via des moyens informatiques, en liaison avec les instances représentatives du personnel et la DRH1 en conformité avec les prescriptions diffusées notamment par la CNIL (dans le domaine de la « cyber-surveillance »). D'un point de vue plus technique, la PSSI peut également se trouver déclinée dans les principaux domaines du système d'information pour détailler les règles de protection : du réseau, des systèmes d'exploitation, d'un SGBD, des serveurs HTTP, etc.
- Guides de configuration ou de recette sécurité : Pour une mise en place efficace, ces règles de protection doivent par contre pouvoir être précisément décrites dans le cas de certains systèmes d'exploitation, certains équipements réseaux ou certains logiciels. C'est alors le rôle des documents opérationnels. Ceux-ci peuvent prendre la forme de guides de configuration ou de cahiers de recette SSI. La principale distinction entre les deux documents tient avant tout à leur mode de mise en œuvre : dans une logique de coopération avec la SSI il peut s'agir d'aider les administrateurs à mettre en place les mesures de sécurité décidées dans l'entreprise, dans une logique de validation et de contrôle2 il peut s'agir d'une procédure de recette (tests) permettant d'autoriser formellement l'ouverture d'un service ou d'un système agréé du point de vue de sa sécurité.
- Analyse des risques : En complément des documents de mise en place, la PSSI peut être accompagnée par un document d'analyse des risques qui permet de mieux comprendre la réalité des principaux biens, des menaces identifiées et des risques recensés dans l'entreprise.
- Synthèse/Suivi : Du point de vue du suivi de la sécurité, il est également important de prévoir l'existence de documents permettant de suivre la mise en place des règles de sécurité (et d'éventuelles violations repérées par des audits par exemple), de consolider les alertes de sécurité identifiées par exemple par certains équipements de sécurité et enfin d'offrir une vue synthétique de la configuration effective des règles de sécurité (telle qu'elle est mise en œuvre dans les

équipements de filtrage par exemple).
- Tableau de bord : Enfin, on peut envisager de rassembler un certain nombre d'indicateurs de sécurité au sein d'un tableau de bord de la sécurité. L'objectif de ce tableau de bord est d'offrir à la direction un état de la situation générale de la SSI, dans l'objectif de présenter les effets de la mise en place de la politique de sécurité de l'entreprise ou éventuellement pour susciter cette mise en place.

Ces différents documents sont présentés plus en détail dans la section correspondante.

La SSI des projets informatiques

Vision idéalisée

L'illustration 1 positionne certaines actions relevant de la SSI par rapport à des étapes classiques du cycle de vie d'un projet de développement logiciel. Ces activités viennent compléter les activités naturelles du déroulement du projet. On peut également distinguer plusieurs types de positionnement par rapport aux projets informatiques vis à vis desquels la gestion pratique de la sécurité informatique pourra être très différente.

- Projets SSI
  - Associés à l'infrastructure de sécurité elle-même
  - Jonction avec les autres projets d'infrastructure
- Assistance aux projets
  - Apporter des compétences
  - Intégrer les projets à la démarche sécurité (et vice versa)
  - Clauses contractuelles
- Validation et contrôle des projets
  - Identifier des vulnérabilités et des risques résiduels
  - Accorder des autorisations d'ouverture

Activités concrètes d'un RSSI

Dans la pratique, les activités suivantes dominent largement le quotidien :

- La veille régulière sur les vulnérabilités, notamment par le biais des CERT (www.cert.org).
- Certains activités opérationnelles : paramétrage du firewall, suivi des IDS.
- Les activités de gestion de la SSI dans l'entreprise : animation du comité de sécurité et des groupes de travail SSI.
- Une activité de production de documentation (PSSI, guides, etc.).
- La gestion des échanges avec les organismes extérieurs (mise à disposition de données pour des partenaires, d'applications, déclarations CNIL, etc.).
- La prise en charge ou le suivi des procédures de l'organisation relatives à la SSI : suivi des tests d'intrusion, gestion des autorisations et des habilitations.

Notes

1. Cela pourrait même être obligatoire pour profiter des dernières évolutions de la loi « Informatique et libertés », et notamment des facilités offertes par la nomination d'un « correspondant à la protection des données à caractère personnel » (les détails restant à préciser).

Sécurité informatique/Le domaine SSI/Documents SSI

Politique de sécurité

La politique de sécurité du système informatique est de plus en plus associée à l'acronyme PSSI, pour « Politique de Sécurité du Système d'Information ».

La structure générale d'une politique de sécurité peut aborder les points suivants :

- Organisation et responsabilités : La PSSI précise l'organisation des fonctions chargées de la sécurité au sein de l'entreprise (postes, rattachements, répartition géographique, cumul, etc.) ainsi que les prérogatives associées à ces fonctions (conduite d'audit, ouverture des services, attribution des droits, gestion des habilitations, etc.).
- Intégration et interactions de la SSI : La PSSI doit également prévoir les modalités d'intégration des fonctions SSI dans l'entreprise et notamment :
  - la manière dont la SSI est prise en compte dans les projets menés par l'entreprise (notamment les projets de développement de logiciels s'il y en a) ainsi que dans les choix techniques effectués (sélection de logiciels, etc.) ;
  - et la manière dont la SSI interagit avec les services chargés de l'exploitation des systèmes informatiques (priorités, indépendance ou non, acquisition des matériels, budget, etc.).
- Objectifs de sécurité : La PSSI doit définir les objectifs de sécurité de haut niveau de l'entreprise. Par exemple, c'est à ce niveau que peut être imposé l'utilisation de systèmes d'authentification à deux facteurs, la nécessité de l'agrément sécurité des serveurs pour certains domaines d'activité, la prédominance de la disponibilité sur les autres aspects de la sécurité (ou l'inverse - ce qui est quand même plus rare), etc. Les objectifs de sécurité, validés par la direction générale, révèlent les intentions de l'ensemble de l'entreprise en terme de sécurité informatique et légitiment les efforts concrets de mise en place. (C'est notamment en ce sens que la PSSI est un document « politique ».)
- Règles générales de sécurité : La PSSI doit non seulement identifier les objectifs assignés, mais également les règles de sécurité générales qu'elles imposent, parmi lesquelles on retrouve certains points récurrents : l'attribution d'un identifiant aux employés, la gestion de leurs habilitations, les règles de rattachement au réseau, la contractualisation des règles avec des partenaires extérieurs. Mais on peut également définir à ce niveau des règles spécifiques : la délégation de certains droits, les autorisations d'ouverture de services réseau, le type des systèmes d'authentification autorisés, la nationalité des fournisseurs, la gestion des obligations légales (traitement de données personnelles notamment), etc.
- Gestion des risques : Les objectifs de sécurité correspondent à des décisions volontaires, mais celles-ci sont bien entendu motivées par les risques encourus par l'entreprise. Idéalement, les objectifs de sécurité doivent correspondre aux mesures permettant de limiter tous les risques majeurs associés à des défaillances de sécurité du système d'information. Mais des risques résiduels existent généralement et la PSSI peut aborder le sujet de la gestion des risques notamment si des efforts d'analyse des risques ou d'audit interne sont menés dans l'entreprise (c'est peut-être déjà le cas, notamment vis à vis du risque financier).

Par rapport à cette structure, la PSSI peut aborder un certain nombre de thèmes correspondant aux principaux domaines techniques du système informatique et du système d'information qu'il me en œuvre. On y recense notamment les thèmes suivants :

- la protection des communications (informatiques mais aussi téléphoniques) ;
- la gestion des violations (blocage, arrêt, correction, suivi, voire sanction) ;
- les interactions avec le domaine de la vie privée - régi en France par la loi de protection des traitements de données à caractère personnel ;
- les procédures de choix et d'achats de matériels ;
- la gestion de la messagerie, notamment si celle-ci est utilisée dans des cas où l'entreprise peut se trouver engagée (par exemple vis à vis d'un sous-traitant) ;
- les procédures de maintenance et d'intervention sur les systèmes en exploitation ;
- les modalités d'enquête et de contrôle de la sécurité ;
- les règles d'identification employées dans le système d'information (les employés permanents constituent le cas le plus simple ; il est loin d'être le seul : intermittents, délégataires, machines, sous-traitants, partenaires, etc.) ;
- les systèmes d'authentification associés à la SSI ;
- les moyens de surveillance mise en place ;
- les systèmes de contrôle d'accès utilisables ;
- la manière dont les contraintes de disponibilité doivent être prises en compte ;
- les règles de gestion du réseau du point de vue de la sécurité (par exemple, point d'accès unique, etc.) ;
- etc.

Selon nous, les caractéristiques d'une PSSI de bonne qualité sont les suivantes :

- Les objectifs et les règles énoncées doivent être réalistes. Il est inutile de prescrire des obligations ou des interdictions qui gênent tellement le fonctionnement des systèmes que les utilisateurs seront obligées de les contourner pour mener à bien leur mission.
- La PSSI doit être applicable, avec des moyens nécessairement limités (notamment du point de vue humain). En général, ceci impose d'accepter certains compromis de réalisation, et même certaines vulnérabilités.
- La politique doit correspondre à une vision à long terme. Ce type de document ne peut pas être révisé tous les ans. Il doit donc être suffisamment générique pour rester en application quelques années. Les détails sont à préciser dans des documents dérivés.
- La clarté et la concision sont nécessaires à certains moments pour énoncer des règles claires. (En général, celles-ci nécessitent toutefois plusieurs paragraphes d'explication pour être bien comprises, notamment dans différents contextes.)
- La PSSI (et notamment ses règles) doit être basée sur des rôles ou des profils d'utilisateurs : les systèmes changent, la notion même d'utilisateur (au sens informatique) peut changer pour des raisons techniques, il faut s'appuyer des notions un peu plus abstraites pour définir les règles de sécurité impliquant les droits des utilisateurs.
- La PSSI doit permettre une définition claire des domaines de responsabilité et d'autorité, notamment sur les systèmes techniques. L'objectif est alors de pouvoir trancher efficacement entre des points de vue contradictoires (ce qui, dans ce domaine technique, est très fréquent).
- La PSSI doit être à jour (elle doit être revue périodiquement ou quand les évolutions de l'entreprise le nécessitent). C'est probablement assez difficile à assurer.

A notre sens, la PSSI doit être communiquée à tout le personnel pour lui permettre de comprendre dans le détail l'impact de la SSI dans son entreprise et la manière dont il a été décidé de la gérer. Cette diffusion de la PSSI peut parfois être plus difficile à réaliser, notamment si les objectifs adoptés négligent explicitement certains risques.

Analyse des risques

Les principales étapes d'une analyse des risques sont les suivantes :

1. Identifier les biens et leur valeur
2. Attribuer des priorités aux biens
3. Déterminer la vulnérabilité aux menaces et les dommages potentiels
4. Attribuer des priorités à l'impact des menaces
5. Sélectionner des mesures de protections rentables

La réalisation d'une analyse des risques apporte des informations très intéressantes pour la définition de la politique de sécurité. Toutefois, de notre point de vue, cette approche masque certaines des décisions qui doivent être prises pour aboutir à la définition de la politique de sécurité : la rentabilité n'est pas un critère suffisant pour décider la mise en place de certaines mesures de sécurité, par ailleurs l'évaluation des menaces et de certains dommages reste assez subjective et rend la plupart des méthodes moins mécaniques qu'elles ne l'avouent.

« Spécifications »

Les spécifications de sécurité peuvent toucher à différents sujets concernant la SSI, avec l'objectif de décrire de manière précise les règles souhaitables et leurs motivations (c'est à dire les informations à protéger) :

- Clauses contractuelles : vis à vis des sous-traitants, des partenaires, etc.
- Réseau : règles d'interconnexion ou d'administration, etc.
- Système : règles d'administration, systèmes utilisables, etc.
- Utilisateurs (finaux, administrateurs, etc.) : charte d'utilisation, etc.
- Collecte des traces : cybersurveillance, protection des données, etc.
- Systèmes d'authentification : protocoles autorisés, etc.
- Relais : modalités de filtrage, surveillance des accès, etc.
- Application (*A*, *B*, *C*, *D*, etc.) : règles spécifiques de gestion pour différents types d'applications (annuaires, données financières, données bureautique, etc.).

Guides de configuration ou de recette

Les guides de configuration ou de recette identifient des points de contrôles :

Ils sont déclinés précisément, par exemple par :

- Système d'exploitation, comme :
  - SunOS 4, AIX 4, 5, Solaris 2.6, 2.7, 2.9, RedHat 6, 7, Debian 2.2, 3.0, OpenBSD 3.5, 3.6, etc.
- Logiciel, comme :
- iPlanet, Apache 2, IIS 6, etc.
- Équipement, comme :
  - Routeurs Cisco 36xx, Switches Cisco Catalyst 7000, 2900, etc. (CatOS ou IOS), Nortell 2430, 5430, etc.

Ils couvrent des éléments de configuration et de gestion concrets, par exemple pour certains paramètres réseau des systèmes d'exploitation suivants :

- Linux procfs

```
echo "0" > /proc/sys/net/ipv4/ip_forward
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

- (Open)BSD sysctl.conf

```
net.inet.ip.forwarding=0
vm.swapencrypt.enable=1
```

- etc.

Documents de mise en service et de suivi

Après contrôle (réussi ou non) de la sécurité d'un système, un document de mise en service identifiant les non-conformités constatées et les vulnérabilités résiduelles du système concrétise l'autorisation de mise en service du système et doit permettre par exemple l'ouverture des accès réseau. Des failles peuvent également être constatées a posteriori par des contrôles de sécurité.

Dans chacun de ces cas, le suivi de la sécurité doit s'appuyer sur des documents identifiant les problèmes résiduels connus et permettant de maintenir ou de faire progresser la sécurité des différents systèmes. On trouve notamment parmi ces documents des matrices de conformité ou des fiches de suivi.

Sécurité informatique/La veille technologique sécurité

La veille technologique, dans le domaine de la sécurité, peut concerner le suivi des nouvelles technologies disponibles sur le marché, mais concerne également le suivi des alertes de sécurité ou plus précisément des nouvelles vulnérabilités découvertes sur les systèmes informatiques. Cette dernière activité de veille est assez particulière au domaine de la SSI, c'est elle sur laquelle nous nous focalisons dans cette section.

Le CERT : un moyen de veille incontournable

Les CERT (Computer Emergency Response Team) sont un des principaux moyens utilisables pour assurer efficacement la veille en sécurité informatique, notamment par le biais du CERT originel, hébergé à l'université américaine de Carnegie Mellon, mais dont les activités opérationnelles ont été transférées début 2004 dans le CERT des États-Unis (US-CERT). Le réseau des CERT a été créé en 1988 en réaction à l'apparition du premier ver majeur sévissant sur Internet. Il s'agit d'un ensemble d'organismes indépendants d'alerte et de consolidation des informations concernant la sécurité des systèmes et des logiciels et les menaces sévissant à un instant donné sur l'ensemble du réseau Internet. La racine de cette organisation est constituée par « le » CERT - en fait le centre de coordination des CERT, le CERT-CC (http://www.cert.org/) - localisé à Carnegie Mellon et dont le site Web, accessible à l'adresse [www.cert.org], est une des principales sources d'information officielle concernant la sécurité informatique au quotidien. Chaque pays et même chaque entreprise est ensuite en mesure de créer ses propres organismes de ce type. Les principaux CERT sont organisés au sein d'un réseau d'échange nommé FIRST (http://www.first.org/) et le réseau accrédite les nouvelles structures souhaitant y contribuer (lesquelles ne sont pas exclusivement des CERT). En France, trois principaux CERT ont vu le jour, avec des succès et des durées de vie variées. Le CERT-RENATER associé au réseau d'enseignement et de recherche (RENATER), le CERTA (http://www.certa.ssi.gouv.fr/) concernant essentiellement les administrations et les services gouvernementaux et le CERT-IST (http://www.cert-ist.com/) dédié à la communauté industrie, services et tertiaire française. Toutefois, dans cette section, nous nous intéresserons avant tout à l'information fournie originellement par le CERT-CC de Carnegie Mellon. Il faut noter que la vision que nous en présentons est probablement désormais obsolète même si l'esprit du fonctionnement d'un CERT quelconque est (ou devrait être) proche de celui du CERT-CC, ne serait-ce que par la généalogie. En effet, l'activité de suivi des vulnérabilités et d'émission d'alertes du CERT-CC a été reprise à compter du début 2004 par l'US-CERT (http://www.us-cert.gov/), structure du DHS (Department of Homeland Security) ministère créé par les États-Unis en 2002 en réaction aux attaques terroristes du 11 septembre de l'année précédente et pour prévenir de nouveaux attentats. Toutefois, l'US-CERT suit le schéma de fonctionnement initial du CERT-CC, et celui-ci continue jusqu'à ce jour à maintenir sa diffusion d'information en s'appuyant sur les données de l'US-CERT.

Vue générale

Page d'accueil du CERT-CC lorsqu'il était en activité

L'activité du jour est résumée dans la synthèse du CERT concernant l'activité quotidienne (voir l'illustration 3). On y fait la distinction entre les alertes de sécurité, les avis de vulnérabilités et le tableau de bord instantanée des principales menaces actives. Les alertes de sécurité sont des documents émis par le CERT concernant des évènements notables du point de vue de la sécurité informatique : vulnérabilité grave et moyen d'y remédier, menace et vulnérabilité associées, etc. L'objectif de ces fiches d'alerte et de fournir un support d'information déjà synthétique et dont le volume reste maîtrisé : le nombre annuel reste de l'ordre de quelques dizaines. Les avis de vulnérabilité sont des fiches d'information technique orientées vers le recensement de toutes les vulnérabilités connues sur les systèmes informatiques à titre de référence. La procédure de diffusion de ces avis est contrôlée : un laps de temps est généralement accordé au constructeur pour proposer un correctif avant la publication de l'avis ou l'avis lui-même est censuré si possible des détails techniques permettant d'exploiter la faille ; toutefois, la politique affichée est de mettre in fine l'information à disposition du public. Enfin, les CERT essaient de recenser le niveau de gravité des différentes menaces actives à un instant donné sur Internet. En effet, les CERT peuvent être destinataires de rapports concernant des intrusions ou des défaillances de sécurité intervenues dans les

organismes avec lesquels ils sont en contact (entreprises, administration). Les différents CERT disposent également de moyens d'observation du réseau IP mondial leur permettant d'identifier les principales attaques utilisées ou les vulnérabilités les plus répandues. En général, une publicité très restreinte est effectuée sur ces éléments techniques. (On peut supposer que cette information est seulement disponible au sein du FIRST.) Mais les CERT peuvent en retirer une vision générale des menaces actives, qu'ils rendent publique.

1) Peut-être même plus particulièrement entre les partenaires américains du FIRST...

Fiches d'alerte

Les principaux éléments d'une fiche d'alerte CERT sont les suivants :

- *Title / Overview* : Titre de l'alerte de sécurité et présentation générale du type d'information fourni par l'alerte (vulnérabilités, menace, etc.).
- *Systems affected* : Identification la plus précise possible des systèmes informatiques concernés par l'alerte (en général, les systèmes d'exploitation).
- *Description* : Une description technique plus détaillée de la ou des vulnérabilités à l'origine de l'émission de l'alerte, orientée vers la protection des systèmes affectés ou la détection d'une tentative d'exploitation.
- *Impact* : L'impact de l'exploitation réussie de la vulnérabilité (prise de contrôle du système, exécution d'un programme avec les privilèges d'un utilisateur normal, déni de service, destruction de fichiers, etc.).
- *Solution* : Les correctifs utilisables sont indiqués dans cette section quand ils sont disponibles, éventuellement accompagnés ou remplacés par des moyens de contournement ou à défaut de détection.
- *References* : Origine de l'alerte, références des avis de vulnérabilités associés et des numéro d'identification de la vulnérabilité (CVE), références des alertes émises par les constructeurs s'il y a lieu.
- *Credit / Vendor Info. / Other Info.* : Informations additionnelles (personnes ayant découvert la vulnérabilité par exemple, remerciements, etc.).

Une fiche de l'US-CERT suit généralement ce schéma. Toutefois, une alerte reste rédigée dans un format relativement libre. C'est surtout l'usage qui a conduit à la structure présentée précédemment, laquelle reste d'ailleurs relativement peu contraignante et permet des niveaux de rédaction assez différents. Il ne faut donc pas voir dans les alertes une base de données au sens strict (pas plus que pour les avis de vulnérabilité d'ailleurs). Il s'agit avant tout d'un moyen de communication, dont le format le plus efficace a été dicté par les usages depuis la création des CERT en 1988 et l'expérience acquise par les équipes de ces organismes. (C'est aussi ce qui peut expliquer pourquoi les CERT les plus anciens sont ceux dont les avis sont généralement de la meilleure qualité.)

Exemples d'avis

- CERT Advisory CA-2003-28 (http://www.cert.org/advisories/CA-2003-28.html) : Cette alerte référence l'existence d'une faille de sécurité appuyée sur un problème classique de débordement de buffer (*buffer overflow*) dans le service Workstation Service de Microsoft Windows. (Cette vulnérabilité a été suivie séparément par le CERT sous l'avis VU#567620 (http://www.kb.cert.org/vuls/id/567620) et a aussi reçu le numéro de référence CVE (http://cve.mitre.org) CAN-2003-0812 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0812).) Les systèmes d'exploitation affectés sont Microsoft Windows 2000 (SP2, SP3, SP4), Windows XP (seul, SP1 et édition 64 bits). L'alerte fait référence au bulletin du constructeur MS03-049 (http://www.microsoft.com/technet/security/bulletin/MS03-049.mspx) et aux différents correctifs disponibles chez le constructeur. Outre l'application des correctifs, les contournements proposés incluent la désactivation du service ou le filtrage des accès réseau utilisés par ce service. L'impact possible est décrit comme permettant d'exécuter des programmes avec les privilèges du système d'exploitation ou des dénis de service ; avec la possibilité que cette faille soit exploitée par un ver. Datée du 11 novembre 2003, l'alerte a été révisée le 20 novembre 2003.

- CERT Advisory CA-2003-26 (http://www.cert.org/advisories/CA-2003-26.html) : Cette alerte référence plusieurs vulnérabilités (six) découvertes dans des implémentations du protocole SSL/TLS (utilisé pour mettre en œuvre HTTPS) et notamment une des implémentations populaires : OpenSSL (http://www.openssl.org/). Dans la plupart des cas, les vulnérabilités concernées conduisent à des dénis de service, mais dans un cas au moins l'exécution à distance de programme semble possible. Les systèmes affectés sont très nombreux. Les versions minimales à utiliser des librairies OpenSSL pour se prémunir de ces problèmes sont indiquées.

- Nous présentons dans une page séparée des séquences d'évènements qui nous ont paru particulièrement intéressantes pour illustrer l'utilisation des avis des CERT.

Bases de vulnérabilités

Les différentes alertes de sécurité générées par un CERT sont complétées par la liste, beaucoup plus détaillée, des avis de vulnérabilités émis pour les différentes failles de sécurité identifiées dans les systèmes informatiques. Toutes les vulnérabilités ne donnent en effet pas lieu à l'émission d'une alerte. Ainsi, par exemple :

- La vulnérabilité CERT VU#567620 est référencée dans l'alerte CA-2003-28 que nous avons déjà mentionnée.
- L'alerte CA-2003-26 est, elle, associée à 6 vulnérabilités différentes.
- L'avis de vulnérabilité CERT VU#936868 (http://www.kb.cert.org/vuls/id/936868) n'a pas donné lieu à l'émission d'une alerte et nous semble toutefois intéressant. D'abord il concerne une faille assez grave, de type buffer overflow, sur un logiciel SGBD très répandu : Oracle. Cet avis est également intéressant car il montre un exemple des limites du fonctionnement par alerte et correctif : la vulnérabilité mentionnée, référencée par Oracle sous le numéro 57, est due au correctif diffusé par ce constructeur suite à une autre vulnérabilité, référencée quelques mois auparavant par Oracle sous le numéro 28.

Les avis de vulnérabilité du CERT ne sont pas les seuls. Des avis constructeurs existent également, et certaines équipes de développement de systèmes d'exploitation gèrent également des avis spécifiques. Par exemple, les avis concernant le système Debian GNU/Linux sont disponibles sur le Web (http://www.debian.org/security/) (l'avis DSA-588 est intéressant par exemple).

Alertes et actions des constructeurs

Sans fournir le même niveau d'indépendance, un certain nombre de constructeurs ou d'éditeurs maintiennent également des équipes chargées du suivi de la sécurité et de la diffusion d'alertes, sous des formes assez variées, par exemple :

- le site sécurité de Microsoft concernant ses logiciels ;
- celui de l'équipe sécurité de Cisco (http://www.cisco.com/go/psirt) ;
- le site sécurité de la distribution ouverte Debian GNU/Linux (http://www.debian.org/security) ;
- la page Web des errata (http://www.openbsd.org/errata.html) d'OpenBSD ;
- etc.

L'information brute

Les constructeurs ou les CERT présentent une information mise en forme, vérifiée et recoupée. Cette information se rencontre aussi ailleurs sur Internet, parfois de manière encore plus anticipée, mais son exploitation demande alors généralement plus d'efforts, plus de compétences, plus de temps et surtout beaucoup plus de sens critique[1]. On peut notamment citer les sources suivantes :

- la mailing list BugTraq, depuis longtemps la source première d'information brute sur la sécurité informatique (archivée à l'adresse http://www.securityfocus.com /archive/ à ce jour) ;
- et plus récemment, des sites dédiés à la sécurité, comme :
  - http://www.securityfocus.com
  - http://www.insecure.org
  - http://www.eeye.com/html/research/advisories/index.html
  - etc.
- et demain... un Wiki peut-être, qui sait?

Notes

1. Ainsi qu'une certaine dose de résistance. *« M$, blah, you deserve to be hacked. »*

Sécurité informatique/La veille technologique sécurité/Scénarios

Les alertes et les avis de sécurité correspondent à la communication de l'information. Mais ils ne permettent pas nécessairement seuls (surtout s'ils sont mal formulés) de comprendre la chronologie et l'impact des problèmes de sécurité sous-jacents. Il est nécessaire de prendre du recul par rapport à l'information qu'ils fournissent pour comprendre comment les utiliser au mieux. Pour cela, nous allons nous intéresser à certains évènements de sécurité ayant eu lieu dans les années précédentes pour retrouver leur trace dans les différents avis de sécurité parus à l'époque. Bien évidemment, a posteriori, la tâche est relativement facile. Il est beaucoup plus utile d'essayer d'anticiper, et d'identifier les avis qui vont avoir un impact important au plus tôt au fur et à mesure de leur arrivée. C'est aussi nettement plus difficile et plus aléatoire même si une vision rétrospective montre bien qu'avec un peu d'exercice, les possibilités d'anticipation existent.

Blaster (été 2003)

Durant l'été 2003, un ver, heureusement non-destructif, s'est rendu célèbre sous le nom de *Blaster*. Du point de vue du CERT-CC (qui, à notre avis, a eu un comportement irréprochable lors de cet événement), les éléments suivants sont à mettre en liaison avec l'histoire de ce problème de sécurité : Une vulnérabilité référencée par l'avis CERT VU#568148 (http://www.kb.cert.org/vuls/id/568148) publié le 16 juillet 2003, signale l'existence d'une faille grave dans le service RPC de la plupart des systèmes d'exploitation de la famille Microsoft Windows.

Compte tenu de la gravité de la vulnérabilité, le CERT émet une alerte de sécurité sous la référence CA-2003-16 le 17 juillet 2003. Cette alerte référence le bulletin de sécurité Microsoft MS03-026, daté du 16 juillet 2003, qui indique notamment les correctifs à appliquer.

Le 31 juillet 2003, le CERT émet une nouvelle alerte référencée CA-2003-19 indiquant qu'il reçoit des rapports concernant des scans variés de recherche de la vulnérabilité référencée précédemment et recense au moins deux techniques d'exploitation de cette vulnérabilité utilisées pour l'exploiter. Le 11 août 2003, le CERT émet une alerte référencée CA-2003-20 pour signaler les premières apparitions d'un ver se propageant rapidement en utilisant la vulnérabilité référencée précédemment. Dans les 3 jours suivant, le CERT précisera l'alerte en question pour indiquer les détails techniques permettant de retirer le ver d'une machine affectée, et de bloquer sa propagation au niveau réseau (en l'absence d'application des correctifs sur les systèmes vulnérables).

Le ver Blaster restera pendant plusieurs mois sur la page du CERT indiquant les menaces actives.

On notera bien évidemment que la prise en compte des informations du CERT de manière préventive pouvait offrir un délai de 4 semaines (ou de 12 jours pour ceux qui ne comprennent pas à la première alerte) pour la mise en place des correctifs de l'éditeur avant l'apparition du ver lui-même. Celui-ci, par contre, utilisant des techniques de propagation relativement efficaces (en tout cas par rapport à ses prédécesseurs) ne laissait guère plus de quelques jours pour réagir une fois sa dissémination entamée. Il est heureux que ce ver n'ait pas eu un caractère destructif.

Netsky (hiver 2004)

Un autre ver, nommé Netsky, particulièrement virulent marqua le début de l'année 2004. Il s'agit en fait d'une série de vers, plusieurs variantes (parfois nommées différemment) ayant été identifiées successivement.

Ce ver est particulièrement représentatif d'une autre catégorie de menaces, utilisant un vecteur de propagation différent : la messagerie électronique. L'exploitation de la négligence des utilisateurs (rendue plus facile par la grande facilité d'activation de programmes dans les interfaces de messagerie) et d'éventuelles failles des clients de messagerie courants permet en effet d'espérer déclencher avec une probabilité assez importante les programmes envoyés via un email d'apparence inoffensif. Pour sa propagation ultérieure, le ver exploite ensuite les informations figurant dans le carnet d'adresse du compte attaqué afin de fabriquer et d'envoyer de nouveaux messages électroniques (généralement falsifiés) contenant des copies du ver. Inauguré par le ver Iloveyou qui consistait en un simple script VisualBasic en 2000, ce type de ver s'est progressivement perfectionné pour améliorer la propagation et rendre la détection plus difficile, tout en incorporant parfois des fonctions plus avancées exploitant aussi des vulnérabilités du système d'exploitation de la machine sur laquelle il s'exécute (scan, installation de services privilégiés, etc.). On a même assisté à une certaine compétition entre les auteurs de différents vers de ce type (Netsky éliminant ainsi un de ses prédécesseurs, nommé Sasser, pour prendre sa place).

La variante Netsky.D dont la propagation a été la plus rapide a illustré un des problèmes lié à la prévention de cette menace via des outils de type logiciels antivirus (de messagerie ou du poste de travail). Entre la détection des premières instances du virus, l'analyse par les équipes des éditeurs de logiciels antivirus, la diffusion d'une signature adaptée, et son déploiement sur des postes de travail dans un réseau d'entreprise de grande taille, il a pu s'écouler environ 8 heures. À notre sens c'est un bon résultat. Le temps de traitement d'un tel événement par cette approche nous semble désormais assez incompressible. Mais pourtant, ce laps de temps a suffit à cette variante d'un virus déjà largement répertorié pour pénétrer dans des réseaux d'entreprises dont les postes étaient pourtant systématiquement équipés de logiciel antivirus. Netsky a donc montré concrètement que la protection offerte par les solutions antivirales agissant sur le principe d'une détection d'attaques connues n'était pas parfaite, même quand les antivirus sont d'une grande qualité technique. C'est bien entendu évident, mais surtout depuis Netsky [1].

ISS/Witty (hiver 2004)

Le mois de mars 2004 a vu également l'apparition d'un ver, baptisé Witty, exploitant une vulnérabilité référencée VU#947254 (http://www.kb.cert.org/vuls/id/947254) par le CERT et spécifique au système d'authentification des logiciels de l'éditeur de logiciels de sécurité ISS (http://www.iss.com/) (*Internet Security Systems*). Le principe de fonctionnement de ce ver est relativement usuel par rapport aux menaces de ce type qui sont survenues à cette période. La portée de Witty est même plus limitée puisqu'il a ciblé les systèmes fournis par un éditeur particulier. C'est sans doute pour cela que Witty n'a pas reçu l'attention large que d'autres vers comme ceux que nous avons mentionnés précédemment ont pu attirer. Malgré tout, Witty marque une rupture dans les menaces observées sur Internet[2].

En effet, ce ver est probablement le premier des codes malveillants à avoir non seulement été conçu pour réussir sa propagation et son exécution à grande échelle ; mais aussi pour effectuer la destruction des systèmes visés. Selon la plupart des informations disponibles, Witty a bien réussi sa mission, en rendant inutilisables les systèmes qu'il a attaqués, parmi lesquels un nombre important de firewall et d'équipements de sécurité (chiffré à une dizaine de milliers). A notre connaissance, le concepteur du ver ou ses motivations n'ont toujours pas été identifiés.

Par contre, ce cas a illustré concrètement les pires scénarios envisageables vis à vis de l'apparition de vers efficaces et destructifs. Witty a visiblement été conçu avec soin et efficacité : il est apparu très peu de temps après la première diffusion publique de la vulnérabilité qu'il exploite (il a donc été préparé avant), sa propagation a été ultra-rapide (estimée à 45 minutes), il a certainement utilisé des techniques de propagation avancées (utilisant un certain nombre de systèmes compromis préalablement pour amorcer plus rapidement la propagation à grande échelle[3]), le code de destruction était simple mais suffisamment sophistiqué pour endommager durablement le système sans pour autant le bloquer immédiatement, le ver était extrêmement compact (700 octets), etc. A notre sens, Witty présente plus de points communs avec une arme efficace qu'avec un programme génial, y compris dans la délimitation de sa cible. Dans tous les cas, Witty est clairement l'œuvre d'un agresseur compétent et déterminé. Celui-ci n'a pas recommencé, mais un ver de ce type utilisant comme vecteur une vulnérabilité affectant des systèmes plus répandus pourrait probablement mettre en danger la majeure partie des systèmes informatiques mondiaux. En tout cas, c'est ce que certains scénarios catastrophes envisagent déjà depuis plusieurs années[4].

Notes

1. Lequel ne faisait pourtant que reprendre le principe d'Iloveyou, menace face à laquelle en 2000 beaucoup d'entreprises avaient justement réagi en généralisant les antivirus de messagerie. Pour l'instant, peu d'entreprises semblent s'intéresser à renforcer sérieusement la sécurité de leur client de messagerie au lieu de jouer au gendarme et au voleur.
2. Voir http://www.schneier.com/crypto-gram-0406.html#9 et http://www.icsi.berkeley.edu/~nweaver/login_witty.txt pour des analyses plus détaillées.
3. Technique dite de *pre-seeding* (pré-amorçage).
4. http://www.icir.org/vern/papers/cdc-usenix-sec02/

Sécurité informatique/Administration et exploitation

Administration

Un certain nombre de difficultés sont fréquemment rencontrées par les équipes d'administration par rapport à la gestion quotidienne de la sécurité (à laquelle ils participent nécessairement) :

- la configuration cohérente de nombreux éléments dans le système informatique ;
- la mise en place des correctifs de sécurité (éventuellement de manière automatique) dont la principale difficulté est d'éviter toute perturbation du fonctionnement normal du système malgré des modifications parfois d'assez bas niveau ;
- le déploiement des mises à jour, lequel peut se faire d'une manière parfois très peu sécurisée par défaut (TFTP, etc.), et qui révèle donc une vulnérabilité de l'exploitation ;
- le besoin de moyens de prise en main à distance pour assurer les tâches d'exploitation :
  - soit via des connexions en ligne de commande comme SSH (ou encore malheureusement Telnet) ;
  - soit par des outils permettant le contrôle d'environnement graphiques complets comme VNC, Patrol, TSE (Terminal Server), Citrix, etc.
- l'identification et le paramétrage de la collecte et du déport des traces du système pertinentes du point de vue de la sécurité (par exemple via syslog),.

Organisation

Fonctions des administrateurs

En général, les administrateurs de systèmes informatiques se répartissent suivant différentes spécialités, reliées aux principaux types de systèmes rencontrés, parmi lesquelles on peut notamment distinguer les fonctions suivantes :

- Administrateur réseau
  - Commutation (LAN)
  - Routage (WAN)
- Administrateur système
  - monde Unix
  - monde MS/Windows
- Administrateur de base de données (DBA)
- Administrateur Web (de plus en plus)
- Administrateurs d'applications (messagerie, GED, etc.)
- Administration des services d'infrastructure
  - DHCP, Active Directory, DNS
  - Systèmes de sauvegarde
- Gestion des postes de travail
  - Configurations types, fabrication
  - Mise à disposition
  - Dépannage, incidents
- et enfin l'administration sécurité

Variété de l'environnement

Les difficultés de répartition des tâches au sein d'une équipe informatique complète s'ajoutent à la grande variété des équipements du système informatique. On peut identifier des systèmes très variés, à la sécurité desquels il faut s'intéresser pour prendre en compte la sécurité du système d'information dans son ensemble :

- Serveurs
- UNIX
- Solaris
- Linux
- RedHat
- Suse
- Debian
- AIX
- Windows
- Novell
- Baies de disques
- Routeurs
- Switches
- PC Windows
- Macintosh
- Robots (sauvegardes)
- Imprimantes
- Boîtiers caches
- Boîtiers firewall
- Éléments logiciels
- Antivirus
- SGBD
- …
- IDS

Correctifs et mises à jours

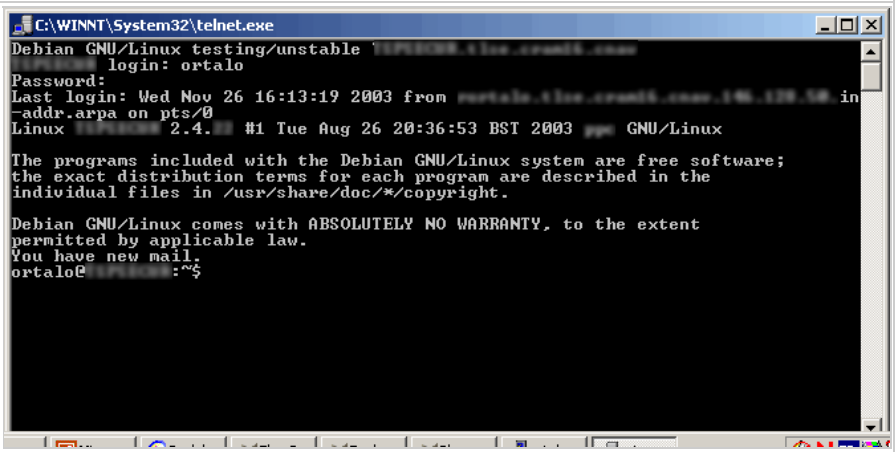
Par rapport à l'application de correctifs sur un système d'exploitation, on peut identifier un certain nombre de préoccupations associées à leur prise en compte.

- La principale contrainte pour l'installation de ces correctifs est de ne pas perturber le fonctionnement normal du système d'exploitation. Dans le cas d'un système bureautique simple, on peut espérer que l'installation n'ait pas d'effets négatifs sur le poste de travail même si, dans certains cas, elle est perceptible pour les utilisateurs (par exemple en rendant certains sites Web moins accessibles). Par contre, quand des applications spécifiques existent sur le système (et c'est fréquemment le cas dans les entreprises pour tous les postes de travail orientés vers une finalité plus précise comme la comptabilité, les achats, la gestion du personnel, etc.), l'installation d'un correctif peut avoir des effets indésirables au point de remettre en cause la faisabilité de cette installation. De plus en plus, l'installation systématique des correctifs est perçue comme indésirable par les équipes d'exploitation (surtout de manière automatique) sans une phase de validation préalable. Celle-ci étant coûteuse, elle remet en cause les mises à jour, surtout en exploitation courante. Dans le cas d'une installation initiale, l'installation des correctifs est plus facile à réaliser, mais elle complique l'installation d'une nouvelle machine.
- La finalité des correctifs sécurité est avant tout de réagir notamment à des alertes de sécurité en corrigeant les failles avant qu'elles aient pu être exploitées. Dans le cas d'une faille effectivement exploitée par une menace active et répandue (comme un ver efficace), l'installation est extrêmement recommandée en l'absence d'autres moyens de protection. Mais dans ce cas, la mise en place doit être rapide. Par contre, tant que la faille n'est pas exploitée, la mise en place du correctif n'est qu'une action de prévention perçue comme optionnelle. Les deux points de vue conduisent bien évidemment à des attitudes contradictoires et difficiles à arbitrer au quotidien, surtout compte tenu du nombre important de correctifs produits par les grands éditeurs.
- Pour gérer les déploiements des correctifs, on dispose de plusieurs catégories de méthodes d'installation :
  - Les patches classiques correspondent à des correctifs à appliquer de manière incrémentale (les uns après les autres) généralement fournis sous forme binaire. Leur taille est variable, souvent petite.
  - A part dans cette catégorie, on peut identifier les correctif fournis pour les sources des logiciels. Diffusés exclusivement dans le domaine du logiciel libre, ces patch sont souvent des deltas (diff) des modifications à appliquer au code source, souvent très petits, dont le déploiement implique la possibilité de reconstruire facilement les exécutables binaires sur le système en exploitation. Dans la pratique, cette méthode de diffusion des correctifs sécurité est surtout envisageable pour les systèmes de la famille BSD ; ou pour les logiciels applicatifs eux-mêmes pris isolément. Par contre, du point de vue de la sécurité, elle présente l'énorme avantage de permettre une compréhension précise de l'erreur de programmation à l'origine de la faille de sécurité, ainsi que du contenu du correctif apporté. Ceci permet aussi de juger de la qualité du correctif (notion très opaque pour les correctifs binaires dans le domaine du logiciel propriétaire).
  - Les systèmes de la famille Microsoft offrent depuis Windows 2000 une infrastructure native d'installation des correctifs, nommée Windows Update ou SUS (pour Software Update Services). Celle-ci, bien que relativement simple, permet notamment aux différents systèmes clients de choisir les correctifs les concernant et de gérer leur ordre d'installation, ainsi que pour les administrateurs de gérer des miroirs des correctifs diffusés par Microsoft (et d'activer ou de bloquer la diffusion de certains patch). Pourtant, SUS reste une solution dont le périmètre est limité, une solution payante plus complexe étant désormais disponible auprès de l'éditeur sous le nom de SMS (Software Management Services) pour gérer plusieurs politiques de diffusion des correctifs, permettre les retours arrières et déployer également des correctifs des logiciels applicatifs (au-delà du système d'exploitation et de ses composants).
  - En effet, la gestion des correctifs de sécurité n'est qu'une instance particulière du problème du déploiement des nouvelles versions des logiciels, marquée par une urgence et des préoccupations un peu différentes (celles de la sécurité). Il est bien évidemment préférable de gérer l'ensemble du système informatique (O.S., applications, configurations, etc.) de manière unifiée par rapport à cette problématique. Les liens entre les différents éléments logiciels et les différents types de machine existants dans une entreprise rendent toutefois ce type de gestion encore très difficilement automatisable (il s'agit en fait pour une grande part du travail d'administration système proprement dit).

Prise en main à distance

En ce qui concerne le problème de la prise en main à distance des équipements, nous mettons en avant un certain nombre de situations courantes.

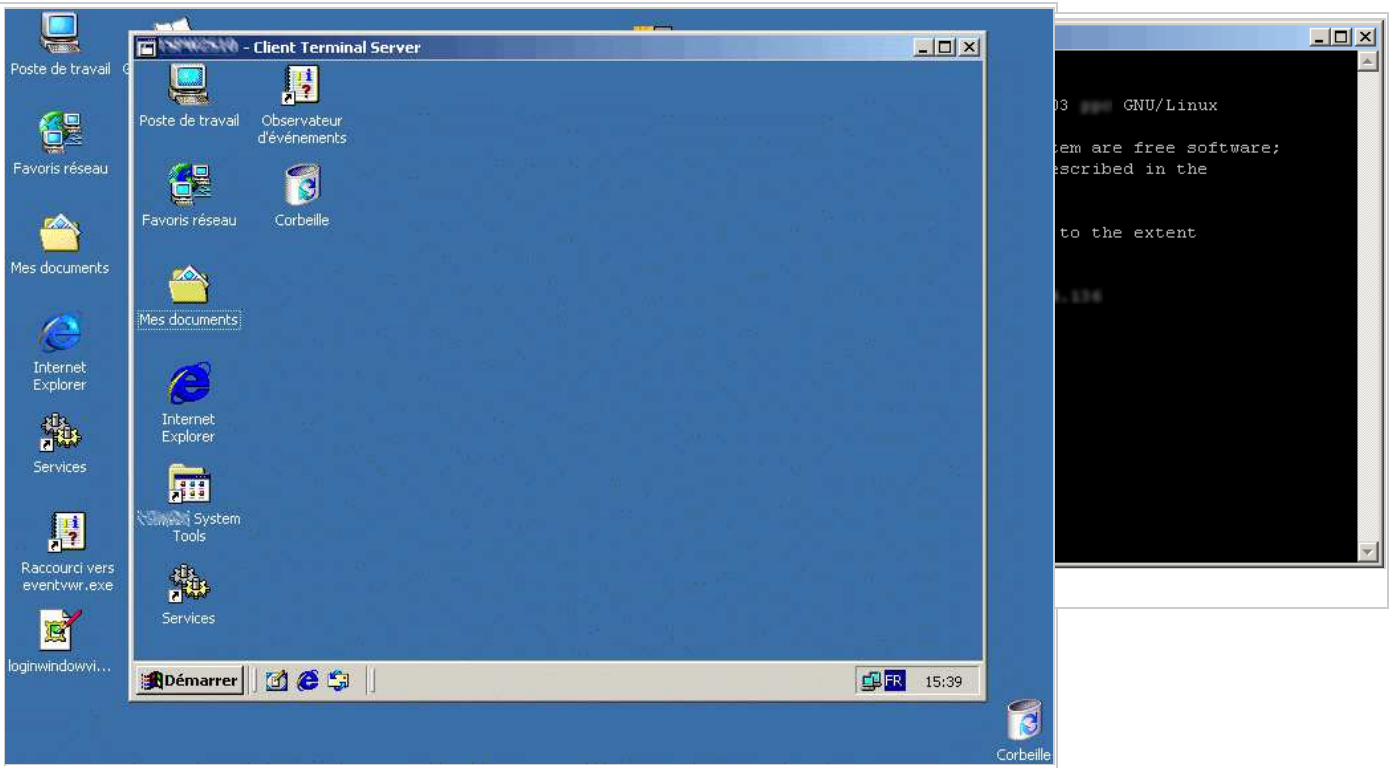
- Dans le domaine de l'administration Unix (largement appuyé sur l'utilisation des interpréteurs de lignes de commandes), une tendance qui trouve son origine dans une problématique de sécurité apparaît désormais clairement pour les connexions à distance : l'opposition entre les protocoles anciens que sont Telnet et RSH, et le protocole fortement protégé OpenSSH. Les figures montrent deux connexions distantes ayant la même origine et la même destination utilisant les deux protocoles. On voit bien que, d'un point de vue fonctionnel, les deux outils sont équivalents. En pratique, peu de choses, à part l'habitude ou les scripts d'administration existants s'opposent désormais à la substitution pure et simple de Telnet et RSH par SSH ou OpenSSH ; que nous recommandons bien évidemment. En fait, dans certains cas, l'utilisation du protocole OpenSSH est même largement plus commode : il gère certaines erreurs de manière plus intuitive, peut propager correctement certains signaux et permet d'effectuer en toute sécurité des transmissions automatiques qui auraient demandé d'inscrire les mots de passe dans les scripts. Le seul argument technique encore valide à notre connaissance pour préférer l'utilisation des anciens protocoles est celui de la connexion vers des systèmes EBCDIC.



Connexion via Telnet (sous Windows 2000)

- De plus en plus d'équipements, notamment des systèmes embarqués, offrent des interfaces d'administration accessibles via HTTP. Dans certains cas, HTTPS est disponible et il faut bien évidemment le préférer à son alternative en clair. Par contre, dans la plupart des cas, HTTPS ne fournit que la protection de la confidentialité du flux. Il serait souhaitable à notre sens, de gérer également l'authentification réciproque de l'équipement et du navigateur d'administration via des certificats comme c'est possible avec HTTPS (en fait avec SSL/TLS), mais c'est encore rarement le cas.

- Dans le domaine de l'administration Windows, même si des moyens d'accès en ligne de commande sont disponibles[1], ils sont rarement suffisants pour permettre d'administrer l'ensemble de la machine. Les principaux moyens d'accès utilisés dans la pratique sont alors ceux indiqués ci-dessous. Ceux-ci offrent généralement une authentification relativement solide (en tout cas pour les versions récentes des protocoles) mais aucune protection du flux TCP sous-jacent.
  - soit le service Terminal Server illustré dans la figure 6, qui fait partie intégrante des systèmes Windows serveurs ;
  - soit, assez couramment, le logiciel VNC (http://www.realvnc.com/) ou certaines de ses alternatives commerciales. (On notera d'ailleurs que la récente variante commerciale de VNC offre justement une palette de fonctions de sécurité et d'authentification nettement élargie par rapport à la version librement disponible.)



Connexion Windows 2000 via TSE (sous Windows 2000)

Systèmes embarqués

De nombreux systèmes pouvant être qualifiés de systèmes « embarqués » sont en fait présent dans le système d'information des entreprises :

- Il s'agit souvent des équipements associés à l'infrastructure réseau (LAN)
- TFTP est largement répandu :
  - Mise à jour des OS embarqués (switch Cisco, PIX)
  - Sauvegarde des configurations
- HTTP et HTTPS également (IHM)
- SNMP est mis en œuvre de manière large mais hétérogène
- SSH apparaît sur les équipements réseau

A ces systèmes embarqués présents en entreprise sont en train de venir s'ajouter des équipements à usage personnel ou à destination des PME. Ces équipements sont parfois désignés sous l'acronyme « SOHO » (Small Office and Home Office). La problématique de leur sécurité est émergente mais avec une diffusion élargie, il faut craindre que son importance ne se révèle directement aux yeux du grand public.

Enfin, la problématique de la sécurité des systèmes embarqués risque de voir son importance s'accroître considérablement avec l'arrivée d'équipements informatiques nouveaux dans le domaine domestique : assistants personnels (PDA), systèmes de réception vidéo entièrement numériques (magnétoscope, récepteur TV satellite ou cable), routeurs ADSL incluant des fonctions de téléphonie ou de télévision, téléphones portables, consoles de jeux vidéo avec accès réseau, tablet PC, assistant de navigation avec récepteurs GPS et pourquoi pas un jour des équipements domotiques (systèmes d'alarmes ou de contrôle d'accès physique informatisés, systèmes de surveillance médicale, voire réfrigérateurs, lave-vaisselle, etc.). Il est également probable que certains de ces équipements incluront des fonctions de sécurité avancées et peut-être très contraignantes pour l'individu (notamment dans le domaine de la diffusion vidéo, ou de la télé-surveillance). Face à ces tendances, le titre de « 1984 » prend le sens que son auteur aurait certainement voulu lui donner : celui d'un effort d'anticipation.

Notes

1. De toute façon, comme il s'agit en général de serveurs Telnet, nous ne les regretterons pas.

Sécurité informatique/Environnement

On peut distinguer différents éléments dans l'environnement des activités de SSI au sein d'un organisme :

- Des entités internes ou associées à l'entreprise :
  - Service études et développements (informatiques)
  - Service exploitation/production (informatique)
  - Sous-traitants
  - Organismes nationaux (éventuellement)
  - Tutelles (éventuellement)
  - DRH et CE/DP (représentants du personnel)
  - Service juridique
  - Cellule chargée des marchés et des assurances (éventuellement)
- Des entités externes, indépendante de l'entreprise :
  - Les services de Justice
  - L'OCLCTIC (http://www.interieur.gouv.fr/rubriques/c/c3_police_nationale/c3312_oclctic)
  - La DCSSI (http://www.ssi.gouv.fr/)
  - La CNIL (http://www.cnil.fr/)
  - Les CERT (http://www.cert.org/)
  - Un CESTI : Centres d'évaluation et de certification de la sécurité d'un système (au sens de l'évaluation de la sécurité selon les Critères Communs, maintenant une norme ISO).
  - L'ENISA (http://www.enisa.eu.int/) : agence européenne consacrée à la sécurité des systèmes d'information créée en 2004.

Nous ne nous attarderons pas sur tous, mais nous vous encourageons à en connaître plus précisément certains accessibles via liens suivants et qui doivent être connus par un acteur du domaine de la sécurité informatique.

La Direction centrale de la sécurité des systèmes d'information (DCSSI) est présentée sur le serveur thématique sur la sécurité des systèmes d'information du gouvernement français, accessible à l'adresse: http://www.ssi.gouv.fr/.

La Commission Nationale de l'Informatique et des Libertés (CNIL) est présentée surson serveur public, accessible à l'adresse: http://www.cnil.fr/.

Sécurité informatique/Eléments de législation

Les références suivantes sont disponibles sur le site officiel de diffusion du droit français http://Legifrance.gouv.fr/.

Textes

Les textes législatifs suivant portent sur les différents thèmes associés à la sécurité des systèmes d'information. Bien entendu, la liste suivante n'est probablement pas exhaustive. (Elle a de plus été établie par un non-juriste, qui espère fortement que l'utilisation d'un media collaboratif permettra à des personnes expérimentées dans ce domaine d'améliorer cette présentation et de la tenir à jour.)

- Protection des données nominatives
  - Loi n°2004-801 du 6 août 2004 relative à la protection des personnes physiques à l'égard des traitements de données à caractère personnel et modifiant la loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés.
  - Directive 95/46/CE du Parlement européen et du Conseil, du 24 octobre 1995, relative à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données.
  - Directive 2002/58/CE du Parlement Européen et du Conseil du 12 juillet 2002 concernant le traitement des données à caractère personnel et la protection de la vie privée dans le secteur des communications électroniques (« Directive vie privée et communications électroniques »).
- Commerce électronique
  - Directive n°2000/31/CE du Parlement Européen et du Conseil du 8 juin 2000 relative à certains aspects juridiques des services de la société de l'information, et notamment du commerce électronique, dans le marché intérieur (« Directive sur le commerce électronique »).
  - Loi n°2004-575 du 21 juin 2004 pour la confiance dans l'économie numérique.
  - Loi n°2004-669 du 9 juillet 2004 relative aux communications électroniques et aux services de communication audiovisuelle.
- Concernant le chiffrement civil ou militaire
  - Loi nº 2001-1062 du 15 novembre 2001, Loi relative à la sécurité quotidienne, art. 30 et art. 31.
  - Loi n°2004-575 du 21 juin 2004 pour la confiance dans l'économie numérique : titre III, chapitre 1er, art. 29 à 40.
  - Décret-loi du 18 avril 1939 fixant le régime des matériels de guerre, armes et munitions (JO 13-06-1939 p. 7463-7466, Rectif. : JO 17-06-1939 p. 7631, Rectif. JO 14-07-1939 p. 8959, Rectif. JO 19-07-1939 p. 9142).
- Signature électronique
  - Loi n°2000-230 du 13 mars 2000, portant adaptation du droit de la preuve aux technologies de l'information et relative à la signature électronique.
  - Directive n°1999/93/CE du Parlement européen et du Conseil du 13 décembre 1999 sur un cadre communautaire pour les signatures électroniques.
- Création de la DCSSI et certification
  - Décret n° 2001-693 du 31 juillet 2001 créant au secrétariat général de la défense nationale une direction centrale de la sécurité des systèmes d'information, J.O. du 02/08/2001, pages : 12496-12497.
  - Décret n° 2002-535 du 18 avril 2002 relatif à l'évaluation et à la certification de la sécurité offerte par les produits et les systèmes des technologies de l'information, J.O. du 19/04/2002, pages : 6944-6946.
- Propriété intellectuelle des logiciels
  - Loi n° 94-361 du 10 mai 1994, loi portant mise en oeuvre de la directive (C.E.E.) n° 91-250 du Conseil des communautés européennes en date du 14 mai 1991 concernant la protection juridique des programmes d'ordinateur et modifiant le code de la propriété intellectuelle.
  - Loi n° 98-536 du 1° juillet 1998, loi portant transposition dans le code de la propriété intellectuelle de la directive 96/9/CE du Parlement européen et du Conseil, du 11 mars 1996, concernant la protection juridique des bases de données.

Codification

Et ces textes sont codifiés de la manière suivante :

1. Code pénal
   1. Sanction pénale des atteintes aux droits de la personne résultant des fichiers ou des traitements informatiques : code pénal, art. 226-16 et suivant(s).
   2. Sanction pénale des atteintes aux systèmes de traitement automatisé de données : code pénal, art. 323-1 et suivant(s).
   3. Sanction pénale du refus de remettre la convention secrète de déchiffrement d'un moyen de cryptologie susceptible d'avoir été utilisé pour commettre une infraction : code pénal, art. 434-15-2 .
2. Code de procédure pénale
   1. Possibilité, pour les officiers de police judiciaire, de procéder à des perquisitions dans les systèmes informatiques et d'avoir accès aux informations contenues dans ces systèmes : code de procédure pénale, art. 57-1, 60-2, 76-3, 77-1-2, 97-1 et 99-4.
   2. Possibilité, pour les officiers de police judiciaire, d'avoir accès aux informations contenues dans les traitements automatisés d'informations nominatives : code de procédure pénale, art. 60-1.
   3. Inclusion des données informatiques dans la liste des pièces susceptibles d'être saisies lors des perquisitions réalisées en flagrant délit ou au cours d'une instruction : code de procédure pénale, art. 56, 94 et 97.
   4. Possibilité, dans le cadre de la lutte contre le terrorisme, pour les magistrats saisis d'une affaire, d'ordonner le déchiffrement des messages cryptés : code de procédure pénale, art. 230-1 et suivant(s).
   5. Conditions d'autorisation et de mise en oeuvre des interceptions de correspondances émises par la voie des communications électroniques : code de procédure pénale, art. 100 et suivant(s).
   6. Conditions d'autorisation et de mise en oeuvre des interceptions de correspondances émises par la voie des communications électroniques effectuées au titre de la procédure applicable à la criminalité et à la délinquance organisées : code de procédure pénale, art. 706-95.
3. Code civil et nouveau code de procédure civile
   1. Valeur juridique et force probante de la signature électronique ou de l'écrit sur support électronique : code civil, art. 1316 et suivant(s); nouveau code de procédure civile, art. 287 et 288-1.
4. Code des postes et des communications électroniques
   1. Conditions d'engagement de la responsabilité civile et pénale des prestataires techniques de l'Internet : code des postes et des communications électroniques, art. L. 32-3-3 et suivants.
   2. Protection de la vie privée des utilisateurs de réseaux et services de communications électroniques et mesures de lutte contre le terrorisme applicables aux opérateurs de communications électroniques, art. L. 34-1 et suivants, art. L. 39-3.
   3. Dispositions générales régissant les communications électroniques et les réseaux de communications électroniques, art. L. 32 et suivants, art. R. 9 et suivants, art. R. 10 et suivants, art. R. 11.
   4. Organisation des règles d'attribution et de gestion des noms de domaines sur l'Internet : code des postes et des communications électroniques, art. L. 45.
   5. Désignation des membres, attributions et fonctionnement de la commission supérieure du service public des postes et des communications électroniques, art. D. 96-1 et suivants.

5. Code de la propriété intellectuelle
    1. Loi n°94-361 portant mise en oeuvre de la directive (C.E.E.) n° 91-250 du Conseil des communautés européennes en date du 14 mai 1991 concernant la protection juridique des programmes d'ordinateur et modifiant le code de la propriété intellectuelle.
    2. Application de la procédure de saisie-contrefaçon aux services de communication publique en ligne portant atteinte à l'un des droits de l'auteur : code de la propriété intellectuelle, art. L. 332-1 (4°).
    3. Protection des droits des auteurs de logiciels : code de la propriété intellectuelle, art. L. 112-2, L. 113-9, L. 121-7, L. 122-6 et suivant(s), L. 131-4, L. 132-34, L. 335-3, R. 132-8 et s. et R. 335-2.
6. Code de la santé publique
    1. Dispositions spécifiques aux traitements automatisés de données de santé à caractère personnel : code de la santé publique, art. L. 1111-8 et L. 1115-1.
    2. Interdiction d'utiliser à des fins commerciales des informations médicales nominatives : code de la santé publique, art. L. 4113-7.
7. Code de la consommation
    1. Contenu et procédure de modification des contrats de services de communications électroniques souscrits par un consommateur : code de la consommation, art. L. 121-83 et suivant(s).
8. Code de la sécurité sociale
    1. Arrêtés d’autorisation de gestion de données nominatives ?

Sécurité informatique/Protection réseau et firewall

Le composant privilégié de la protection réseau sur TCP/IP reste le firewall. Sous cette désignation générique, qui cache parfois des technologies extrêmement différentes, on désigne en réalité la mise en place d'un système informatique placé en situation de coupure par rapport aux flux de communication réseau, et capable d'autoriser ou d'interdire ces flux en fonction de son paramétrage. Lequel est généralement appelé la « politique de sécurité » du firewall. C'est une dénomination un peu abusive à notre sens, et nous préférerons parler des règles de filtrage réseau du firewall, ou plus simplement des règles du firewall quand le contexte ne permet pas de confusion.

Bien que formellement un niveau de sécurité identique (voire supérieur) puisse être atteint avec d'autres approches[1], la mise en place d'un firewall reste une voie largement répandue pour tenter d'améliorer la sécurité d'un système informatique et de son réseau. En effet, dans un système informatique pré-existant, ce type d'équipement présente l'avantage d'être indépendant des systèmes informatiques déjà présents, qu'il s'agisse de serveurs d'applications, de serveurs de fichiers, ou d'équipements d'interconnexion réseaux. Cette autonomie est vérifiée à la fois du point de vue technique mais aussi du point de vue de l'administration et donc par rapport à l'organisation d'un service informatique (où l'intégration d'un périmètre de responsabilité et de compétence nouveau n'est pas forcément facile). Par ailleurs, bien qu'il soit assez intrusif (il faut qu'un firewall soit en coupure des flux réseaux pour remplir sa fonction) et parfois difficile à paramétrer quand les flux sont nombreux (notamment dans le cas d'un réseau d'entreprise segmenté en interne) ce type d'équipement se positionne assez naturellement dans le réseau au niveau des interconnexions entre organismes, entre sites, ou entre une entreprise et les réseaux externes (généralement Internet). Quoiqu'elle fasse rarement l'objet d'une réelle justification en terme de besoins de sécurité[2], cette mise en place en « entrée » et en « sortie » du réseau est facile à proposer et à appréhender, notamment par les décideurs. Enfin, malgré tout, par rapport à une situation antérieure où la sécurité n'est absolument pas prise en compte et notamment par rapport à des applications fermées, héritées et figées (ce qui ne veut pas forcément dire qu'elles soient anciennes), la mise en place d'un firewall reste une réelle opportunité d'obtenir le minimum de contrôle et de protection qui est désormais nécessaire pour un système d'information professionnel.

Par contre, du point de vue de la mise en oeuvre technique, un firewall est un équipement assez intéressant, touchant à la fois au fonctionnement protocolaire d'IP, de TCP, et d'UDP, ainsi qu'éventuellement au fonctionnement des applications. Même en se limitant aux protocoles principaux (TCP et UDP), la réalisation efficace du filtrage réseau, à la fois en terme de performance, de facilité d'administration et de sécurité, implique des fonctions assez avancées comme le suivi de l'état des connexions TCP par exemple. Par ailleurs, ces systèmes offrent un certain nombre d'opportunités d'actions nouvelles dont certaines, comme la translation d'adresses, sont parfois cruciales pour offrir un accès réseau à un grand nombre d'utilisateurs, et d'autres comme la gestion de la qualité de service au niveau TCP sont techniquement très intéressantes pour améliorer la qualité des services réseaux.

Principaux modes de fonctionnement

Les premiers firewall ont débutés en offrant simplement des capacités de filtrage basées sur les adresses source et destination présentes dans chaque paquet IP, en complément des fonctions de la pile IP d'un système d'exploitation et notamment du routage. A l'heure actuelle ces fonctions, ne suffisent plus pour qualifier réellement un système informatique de firewall. Ces « filtres IP », quoique parfois particulièrement utiles[3], ne permettent pas de remplir le rôle de contrôle que l'on attend d'un système firewall complet. La confusion existe encore toujours parfois (par exemple, le « firewall personnel » intégré dans Windows XP semble bien être un simple moteur de filtrage IP) mais nous nous intéresserons dans ce chapitre à des logiciels plus avancés, capables d'effectuer des décisions de filtrage ne se limitant pas à l'examen de l'en-tête de chaque paquet IP pris isolément.

En excluant ces systèmes, deux grandes catégories de firewall restent à distinguer : les firewall avec suivi d'état, et les firewall basés sur des proxy. Ces catégories ne sont pas exclusives et les implémentations les plus sophistiquées mélangent parfois des idées provenant des deux approches.

Firewall avec suivi d'état

Les firewall avec suivi d'état sont capables de suivre les différentes étapes de la mise en place d'une communication réseau, et notamment toute la séquence d'une communication TCP : l'échange initial (SYN, SYN+ACK, ACK), les paquets de données (PSH, ACK) et la terminaison (RST, RST+ACK). Dans le cas d'UDP, qui reste normalement un protocole non connecté, le suivi d'état s'appuie généralement sur l'hypothèse d'un fonctionnement des applications en mode « question/réponse » dans lequel un message UDP initial est suivi d'un deuxième message en sens inverse contenant la réponse.

La prise en compte du fonctionnement normal de ces sessions TCP (ou des pseudo-sessions UDP) doit inclure également la prise en compte des erreurs de fonctionnement, comme l'émission éventuelle d'un paquet ICMP indiquant l'échec d'une tentative de connexion TCP ou de l'acheminement d'un message UDP. L'objectif du suivi d'état de ces firewall est d'utiliser l'ensemble des informations collectées sur le déroulement des sessions de communication afin d'autoriser seulement les paquets IP appartenant à une session à transiter sur le réseau, et de protéger au mieux cette communication en détectant (et en bloquant) tout fonctionnement anormal par rapport aux définitions protocolaires ou aux flux réseau usuels.

Le suivi d'état permet également de faciliter la définition des règles de filtrage en permettant une autorisation implicite des différents types de paquets IP associés à une communication : ainsi il n'est pas nécessaire de prévoir les autorisations IP bi-directionnelles nécessaires à la réponse à une demande UDP, ou au déroulement d'une communication TCP. La formulation des règles de filtrage est alors beaucoup plus naturelle : on autorise en quelque sorte seulement le premier paquet initiant une communication, les autres autorisations nécessaires étant automatiquement dérivés à l'aide des informations de suivi d'état.

Ce fonctionnement est également plus performant dans la pratique : en effet, les informations de suivi d'état sont associées aux communications actives. Les autorisations issues des tables d'état sont donc normalement relatives à la grande majorité des paquets IP manipulés à un instant t par la pile réseau. Ce sont donc celles qui peuvent faire l'objet d'une mise en oeuvre prioritaire et particulièrement optimisée de manière à améliorer la performance de l'ensemble du firewall. Le parcours systématique de toutes les règles de filtrage du firewall (qui peuvent être extrêmement nombreuses si la configuration est très précise) n'est alors plus nécessaire que pour une minorité de paquets IP, ceux associés à des débuts de communication. Dans la pratique, ce mode de fonctionnement est fréquemment optimal et cette optimisation n'exclut que des types de trafic réseau quasiment pathologiques (comme des connexions extrêmement fréquentes et de très courte durée)[4]. Les firewall avec suivi d'état peuvent donc généralement supporter un nombre de règles de filtrage largement supérieur aux filtres IP (pourtant techniquement plus simples). Les dernières générations de firewall, notamment certaines visant à fournir des capacités de filtrage sur des réseaux Gigabit, mettent en œuvre ces autorisation basées sur les tables d'état directement au niveau hardware, dans des ASIC dédiés. La partie logicielle du firewall n'est alors sollicitée que pour l'examen des connexions nouvelles.

Firewall proxy

Face aux firewall à suivi d'état, qui ont connu une diffusion importante, l'apparition de composants de plus en plus sophistiqués (modules embarqués dans les noyaux des systèmes d'exploitation, logiciels influant sur le fonctionnement des piles réseau, voire mise en œuvre matérielle par des composants associés aux cartes réseaux) et une débauche d'investissement, les firewall basés sur les logiciels proxy peuvent parfois donner l'impression qu'ils ne sont pas en mesure de soutenir la compétition. Pourtant, à l'origine, cette approche de la mise en œuvre du filtrage des communications réseau visait avant tout à offrir le meilleur niveau de sécurité en envisageant de réexaminer, pour chaque application souhaitant communiquer au travers du firewall, le contenu du flux de communication, *en tenant compte de la sémantique applicative*.

Pour chaque type d'application, il faut alors disposer d'un logiciel proxy (relais) capable d'analyser dans une certaine mesure le flux de communication afin de le

contrôler précisément (sans aller jusqu'à pouvoir l'exécuter, il faut que le proxy puisse comprendre le sens de la communication qu'il relaye), voire de n'autoriser que certaines des actions de communication possibles. Il est donc nécessaire de développer des relais spécifiques pour chaque application que l'on souhaite autoriser. En règle générale, on peut espérer disposer de relais concernant les principaux protocoles de communication utilisés :

- HTTP/HTTPS : pour la navigation Internet (avec une problématique plus particulière associée au chiffrement d'HTTPS dont le « relayage » peut impliquer un déchiffrement/rechiffrement).
- FTP : pour les échanges de fichiers et le contrôle des commandes admises (PUT, GET, DEL, etc.), ainsi que pour la prise en compte des différents modes de fonctionnement (actif, passif).
- Telnet : pour le contrôle éventuel du contenu de la session, mais surtout pour assurer une authentification additionnelle.
- X11 : pour éviter les débordements possibles via le protocole X11 tout en permettant un affichage à distance de qualité acceptable.
- SOCKS : pour réaliser un relayage « générique » de tout protocole TCP/UDP en ajoutant une authentification en sortie (si possible transparente).
- H.323 : pour aborder un protocole basé sur UDP mais n'utilisant absolument pas un mode de fonctionnement demande/réponse tout en ayant de fortes exigences de continuité du flux réseau (transport de la voix).

Cette approche, lourde en terme de développement, est, on le sent bien, également parfois problématique en terme de performance : le logiciel proxy effectue une analyse de type applicatif *en supplément* de celle effectuée sur la machine réellement destinataire.

Pourtant, malgré ces inconvénients, cette approche présente un certain nombre d'avantages, qui font que l'utilisation des proxy reste toujours incontournable dans de nombreuses configurations de filtrage réseau. D'abord, le développement d'un proxy, meme s'il est limité à un seul type d'application, est généralement plus facile que la réalisation d'un module de système d'exploitation, s'exécutant sans protection mémoire dans un environnement très contraignant. Par ailleurs, les proxy, parfois en complément d'un firewall avec suivi d'état mais aussi parfois tout à fait isolément, sont nécessaires pour réaliser facilement certaines fonctions, comme le filtrage des commandes applicatives pour FTP, le filtrage d'URL dans le cas d'HTTP, la lutte contre le spam avec SMTP ou l'ajout d'une authentification transparente sur un relais générique. A l'inverse, certains firewall s'exécutant prioritairement en mode noyau en association avec un système de suivi d'état peuvent intégrer des interfaces modulaires visant justement à faciliter la mise en œuvre de règles de filtrage applicatif (c'est notamment le cas de Netfilter sous Linux).

En règle générale, les deux approches sont complémentaires. L'utilisation d'un relais peut s'avérer assez naturelle si l'application concernée intègre la notion de relayage dans son fonctionnement (c'est par exemple le cas avec SMTP) et si le système d'exploitation hôte, notamment son module de filtrage noyau, facilite la mise en oeuvre du proxy. Le firewall dans son ensemble en est alors largement bénéficiaire. C'est une approche pragmatique qu'illustre bien OpenBSD : même si le firewall avec suivi d'état du noyau est un composant essentiel, à l'usage, l'importance des proxy disponibles se révèle rapidement pour certaines des problématiques mentionnées précédemment, et la coopération entre les deux éléments, quand elle permet le relayage transparent (transparent proxying) offre un réel confort aux utilisateurs finaux. (Ce qui n'est pas inutile pour faire passer l'ajout d'une barrière de sécurité additionnelle.)

Équipements et solutions disponibles

Solutions commerciales

La liste suivante tente de rassembler les principales solutions commerciales pertinentes au moment de la rédaction de cette section.

- Leaders
  - Firewall-1 (CheckPoint)
  - PIX (Cisco)
- Challengers
  - Netscreen
  - Cyberguard
  - ISA (Microsoft)
  - IOS FW (Cisco)
  - Sidewinder
  - SonicWall
  - WatchGuard
  - ...
- Français
  - Netwall (Evidian/Bull)
  - M>Wall (Matranet)
  - Arkoon
  - Netasq

Solutions open-source

Dans le domaine des logiciels librement disponibles, les principaux systèmes d'exploitation existants incluent également des composants logiciels susceptibles de jouer le rôle de firewall. Ceux-ci ont évolués progressivement, mais constituent à l'heure actuelle des mise en oeuvre complètes et généralement très efficaces.

- OpenBSD pf (http://www.benzedrine.cx/pf.html) : Le firewall disponible avec le système d'exploitation OpenBSD constitue probablement une des implémentations les plus modernes d'un firewall complet. Elle a été démarrée en 2001 suite à un désaccord des principaux auteurs d'OpenBSD avec l'auteur du firewall utilisé à l'origine dans ce système d'exploitation (jusqu'à la version 3.0), IPFilter (voir ci-dessous). Ce désaccord concernait la licence d'IPFilter et il a conduit les principaux développeurs d'OpenBSD à retirer cette première implémentation exogène de leur distribution standard. Étant donné l'orientation résolument affirmée d'OpenBSD sur les problématiques de sécurité, ce vide a été rapidement comblé (en moins de 6 mois) par une implémentation nouvelle mais très complète d'un firewall à suivi d'état TCP/UDP avec des capacités de translation d'adresses (NAT). Cette implémentation, baptisée pf (pour Packet Filter) a continué à évoluer, en s'intégrant notamment étroitement avec le système de gestion de qualité de service réseau (QoS) ALTQ et en améliorant la facilité de gestion et les performances (avec des fonctions comme les listes d'adresses dynamiques, un logiciel séparé de gestion des traces, ou plus récemment des capacités d'optimisation des règles de filtrage et un protocole de gestion de la redondance). Le langage de définition des règles de filtrage est un langage textuel, dans la droite ligne des langages de configuration Unix, mais très commode (il a visiblement bénéficié de toute l'expérience d'administrateurs réseaux largement accoutumés aux problèmes de la protection et du filtrage). L'ensemble constitue désormais un logiciel très complet auquel il ne manque plus grand chose pour se mesurer aux meilleures implémentations commerciales en terme de fonctionnalités. Trois ans après la mise en oeuvre initiale, pf a désormais été adopté par les cousins FreeBSD, NetBSD ou même DragonFlyBSD.
- Linux/IPTables (Netfilter) (http://www.netfilter.org/) : Le firewall intégré au noyau Linux a connu une évolution plus progressive, évoluant notamment avec la version 2.4 du noyau pour intégrer le suivi d'état (TCP, UDP, ou autre). Il a d'ailleurs changé de nom plusieurs fois, la version la plus aboutie étant baptisée Netfilter (ou IPTables, par abus du nom de l'utilitaire de configuration du noyau iptables(8)). Cette implémentation présente notamment la caractéristique d'une mise en oeuvre extrêmement modulaire permettant la prise en charge de protocoles complexes ou nouveaux par le développement de modules de filtrage venant

s'intégrer à Netfilter dans le noyau Linux.
- Linux/IPChains Linux/ipfwadm : Les versions plus anciennes du firewall Linux étaient baptisées IPChains (noyau 2.2) et ipfwadm (noyau 2.0). Il s'agissait là d'un filtre de paquet n'intégrant pas complètement le suivi d'état des sessions réseaux (notamment pour TCP). Elle fut assez populaire, car associée à une phase de forte diffusion du noyau Linux lui-même et probablement parce qu'elle intégrait des fonctions de translation d'adresses (appelées masquerading dans IPChains) au moment où leur intérêt devenait crucial, mais elle a été totalement remplacée par Netfilter. Elle est ici avant tout mentionnée pour éclaircir la confusion, parfois encore possible, entre le firewall du noyau Linux 2.2 (IPChains) et celui des noyaux ultérieurs à Linux 2.4 (Netfilter/IPTables), ou celle, quand même moins répandue, entre le firewall originel de FreeBSD et NetBSD (ipfw) et celui de Linux 2.0 (ipfwadm).
- IPFilter (http://coombs.anu.edu.au/ipfilter/) : Il est encore important de mentionner IPFilter dans le domaine des implémentations librement disponibles d'un logiciel firewall. Bien que désormais supplantée (notamment en terme de diffusion) par les alternatives issues d'OpenBSD et de Linux, IPFilter constitue encore une des rares mises en œuvre fonctionnant sur une gamme de systèmes d'exploitation, et notamment des Unix commerciaux (Solaris, HP-UX, etc.). C'est un firewall à suivi d'état TCP/UDP très complet, dont le langage de paramétrage a largement inspiré les réalisations plus modernes (notamment pf).
- ipfw : Il s'agit du firewall présent initialement dans les différents membres de la famille de système d'exploitation BSD (FreeBSD et NetBSD notamment). La tendance récente sur ces systèmes est à l'intégration de l'implémentation du firewall pf issu d'OpenBSD en remplacement d'ipfw, cité ici seulement pour mémoire.

Aspects architecturaux

Exemples

Intégration différée pour:

- validation éventuelle de l'utilisation des copies d'écran constructeur (Firewall-1 et PIX) ;
- à rédiger pour les firewall open-source (Linux/Netfilter et OpenBSD/pf prévus).

Authentification utilisateur

L'installation d'un firewall peut permettre d'introduire une authentification additionnelle sur des protocoles n'en comportant pas ou incluant une authentification offrant un niveau de sécurité insuffisant. Cette authentification est ajoutée assez naturellement dans le cas des firewall proxy : c'est alors le proxy qui prend en charge l'authentification d'un utilisateur souhaitant accéder à un service réseau contrôlé par le firewall ; mais elle est également possible avec les firewall fonctionnant au niveau TCP. Dans la plupart des cas, l'utilisation de cette authentification additionnelle implique l'installation sur le poste utilisateur d'un composant logiciel spécifique.

Voici quelques exemples de mise en œuvre de ce genre d'authentification :

- ajout d'une phase d'authentification aux sessions Telnet d'administration (par exemple à destination des machines en DMZ) afin de pallier au problème de la transmission du mot de passe en clair sur les sessions Telnet ;
- authentification et autorisation des accès HTTP via le protocole NetBIOS sur le firewall Microsoft ISA Server – cette authentification est totalement transparente si le navigateur Web utilisé la supporte (c'est bien évidemment le cas pour Internet Explorer sur système d'exploitation Microsoft) et constitue un des principaux avantages pratiques d'ISA Server avec Internet Explorer ;
- authentification transparente des flux réseaux relayés sur le protocole de relais générique SOCKS v5 [RFC 1928] avec des extensions pour la prise en compte de NetBIOS, en se basant sur la couche WinSOCKS existant dans les systèmes d'exploitation Microsoft ;
- authentification forte et protection dans un tunnel IPSEC des accès nomades sur un firewall type CheckPoint VPN-1, en utilisant un module d'accès SecuRemote ou SecureClient sur le poste client ;
- authentification forte des accès via une passerelle OpenBSD en conditionnant l'activation de règles de filtrage réseau à l'ouverture d'une session SSH sur le firewall en utilisant authpf(8) (http://www.openbsd.org/faq/pf/authpf.html) comme login shell.

QoS

Comme nous l'avons déjà mentionné précédemment, la gestion de la qualité de service nous semble s'effectuer très efficacement au niveau de la définition des règles de contrôle du trafic réseau constituant la politique de sécurité du firewall. Cette approche est notamment illustrée dans l'extension logicielle FloodGate-1 de CheckPoint, ainsi que dans l'intégration d'ALTQ avec PF (http://www.openbsd.org/faq/pf/queueing.html) sous OpenBSD permettant d'utiliser les règles de PF pour associer des paquets aux différentes classes de trafic ordonnancées via les stratégies ALTQ. Ce type de fonction permet d'offrir des garanties de bande passante aux utilisateurs, mais aussi par exemple d'utiliser des protocoles interactifs facilement en présence de flux de transfert massifs (type FTP) ou de faire coexister des flux réseaux continus allant dans les deux sens quand une des directions est saturée (ce qui est notamment le cas sur des liaisons asymétriques, type ADSL). Dans certains cas, les améliorations obtenues sont extrêmement importantes (voir http://www.benzedrine.cx/ackpri.html)...

Notes

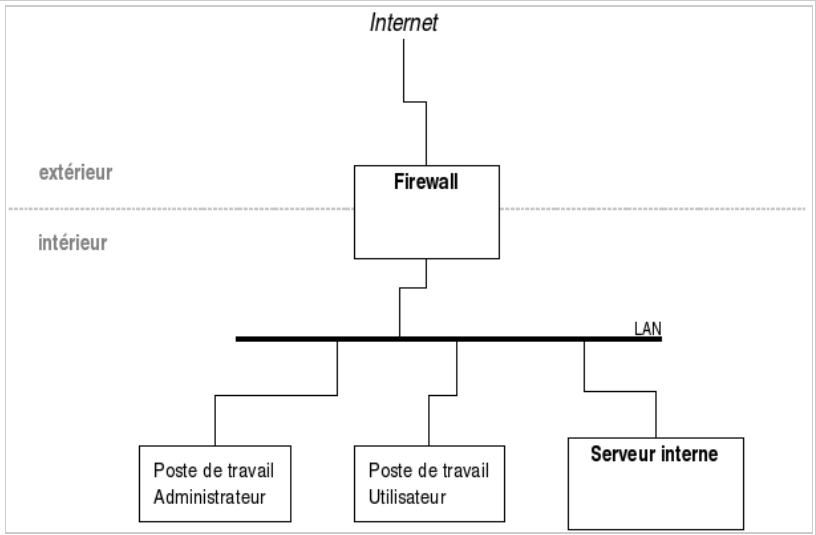
1. Par exemple la certification ou l'agrément des configurations, le confinement des services accessibles par le réseau, l'utilisation de systèmes d'exploitation multiniveau, ou tout simplement l'utilisation d'un système d'exploitation de confiance.
2. Est-il véritablement réaliste de s'imaginer que le principal besoin de sécurité consiste à se protéger de « l'extérieur » dans une entreprise ? Même si c'est le cas, couvre-t'on vraiment l'intégralité de ce besoin avec un firewall, (c'est à dire, entre autres, toutes les possibilités de collusion ou de rebond, même involontaire, avec un élément interne) ?
3. Les filtres IP sont généralement présents par défaut dans le logiciel de la plupart des marques de routeurs et parfois même sur les *switch*, c'est à dire en plein cœur du réseau. Quand la situation est suffisamment grave pour permettre de convaincre l'administrateur réseau ou l'opérateur de se pencher sur leur mise en œuvre, ils offrent un moyen particulièrement rapide et efficace de bloquer des attaques avérées (comme celles associées à la propagation d'un virus par exemple). A contrario, une fois les opérateurs réseaux convaincus, ces filtres s'installent parfois si durablement qu'ils arrivent à trouver leur place dans certains *firmware*, à devenir des arguments commerciaux, et à rendre inutilisables certains numéros de ports (4444/TCP par exemple).
4. Il existe ou il existera certainement des applications ou des configurations pour lesquelles un tel trafic réseau est nécessaire. Il faut donc garder à l'esprit cette optimisation face à d'éventuels problèmes de performance d'un firewall, surtout si celui-ci a de nombreuses règles de filtrage. C'est également à prendre en compte dans le cas d'une mise en œuvre sur réseau très haute performance avec des firewall contenant des ASIC dédiés, qui présentent forcément des limites (peut-être très faibles) en terme de taille des tables d'état hardware, et donc de connexions actives suivies directement par les ASIC.

Sécurité informatique/Protection réseau et firewall/Aspects architecturaux

Principes de fonctionnement

L'utilisation la plus simple d'un firewall consiste à le placer en coupure sur le lien de sortie vers Internet d'un réseau d'entreprise, en le configurant sur le principe d'une « diode ».

Dans cette architecture, présentée dans la figure suivante, le firewall interdit toutes les connexions entrantes en provenance d'Internet, et autorise seulement les connexions sortantes. Il s'agit dans ce cas des connexions TCP ou UDP. Les autres types de paquets IP sont généralement tous rejetés, quel que soit leur direction. Quelques exceptions sont parfois utiles, notamment pour permettre à certains types de paquets ICMP de fonctionner : il est généralement souhaitable d'autoriser certaines des machines du réseau interne à réaliser des « ping » (c'est-à-dire un échange d'un paquet ICMP echo request sortant et de la réponse ICMP echo reply entrante) afin de permettre un test commode de la connectivité avec le réseau Internet.



Filtrage mis en place par un firewall

Les flux réseaux à destination des interfaces du firewall lui-même font généralement l'objet de règles de filtrage plus précises, n'autorisant que les connexions d'administration provenant du réseau interne sur des ports TCP particuliers (comme SSH par exemple). Le firewall lui-même peut être soit extrêmement discret (ne répondant absolument pas aux demandes de connexion non autorisées1, invisible pour des ping de toute provenance) ce qui est parfois malcommode mais est de plus en plus recommandé du côté en contact avec Internet ; soit un peu plus visible (répondant aux demandes de ping2, répondant aux demandes de connexion non-autorisées par des rejets explicites3) ce qui est notamment utile du côté du réseau local pour limiter le temps d'attente d'une connexion refusée.

Les principaux besoins de sécurité auxquels répond cette configuration sont les suivants :

- la protection du réseau local, contenant habituellement des postes de travail ou des serveurs n'offrant pas un niveau de sécurité suffisant pour être directement visibles sur Internet (présence de vulnérabilités connues, de services réseau peu protégés, etc.) ;
- la mise à disposition d'un accès Internet (sortant) grâce à l'utilisation des fonctions de translation d'adresses du firewall, qui permettent à un nombre important de machines du réseau local (typiquement configuré avec un adressage IP privé, non routable) d'accéder à des serveurs publics en utilisant une seule adresse IP publique (généralement celle de l'interface Internet du firewall lui-même) ;
- le contrôle d'accès en sortie, permettant de préciser les postes de travail ou les serveurs disposant d'un accès à Internet dans l'entreprise ;
- et enfin, bien que ce ne soit généralement pas un objectif de sécurité explicite de l'organisation, la protection du réseau Internet lui-même contre des attaques provenant du réseau interne (en contrôlant les flux réseau disponibles pour les machines du réseau local, ce qui permet d'interdire un scan de ports sortant par exemple).

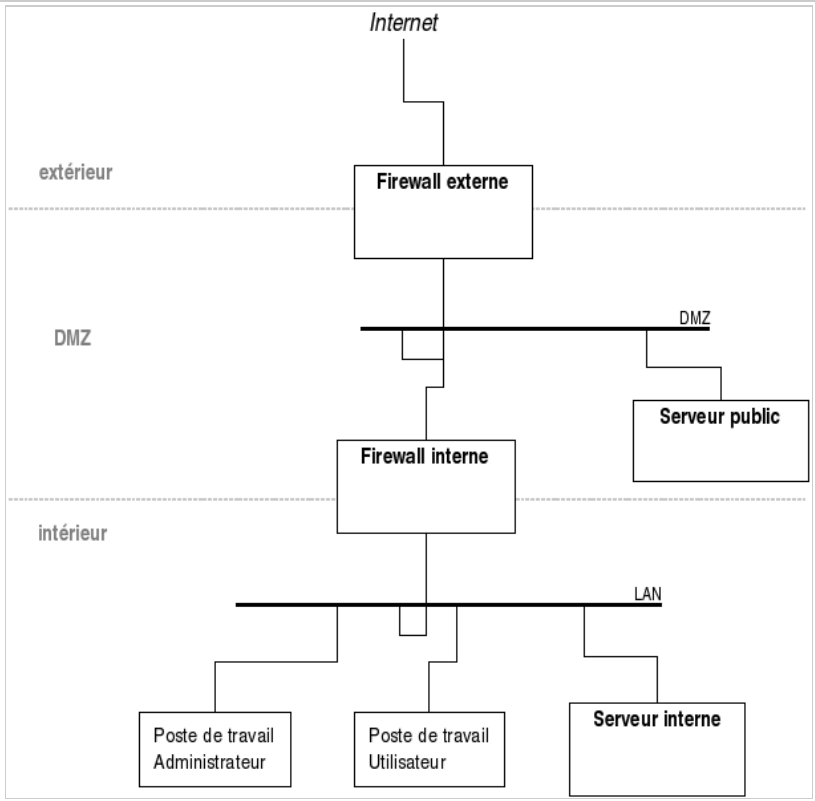
Dans cette architecture, tous les équipements connectés au réseau interne sont placés au même niveau de sécurité du point de vue du firewall. Notamment, les flux réseaux internes au LAN ne sont pas contrôlés.

« Niveaux » de sécurité et DMZ

Dès que l'on souhaite utiliser une connexion à Internet pour des applications plus avancées se fait ressentir le besoin de disposer de plus d'un niveau de sécurité. La connexion du réseau d'entreprise vers Internet peut impliquer également la mise en place d'un certain nombre de services réseaux visibles depuis Internet : serveur HTTP, serveur de messagerie par exemple. Dans ce cas, les besoins associés à ces serveurs se situent dans un niveau de sécurité nouveau : ils ne bénéficient pas du même niveau de protection que les machines du réseau interne (totalement masquées, mais inatteignables), mais il reste généralement souhaitable de ne pas les exposer directement sur Internet et d'assurer leur protection.

Pour répondre à cette problématique, en utilisant un matériel similaire à celui présenté au 3.1.3.1, on peut envisager d'utiliser deux équipements : un firewall interne et un firewall externe. Le firewall interne est configuré en diode et protège le réseau local tout en lui permettant d'accéder à Internet et aux serveurs publics de l'entreprise. Le firewall externe autorise les connexions entrantes à destination des serveurs publics pour que ceux-ci remplissent leur fonction.

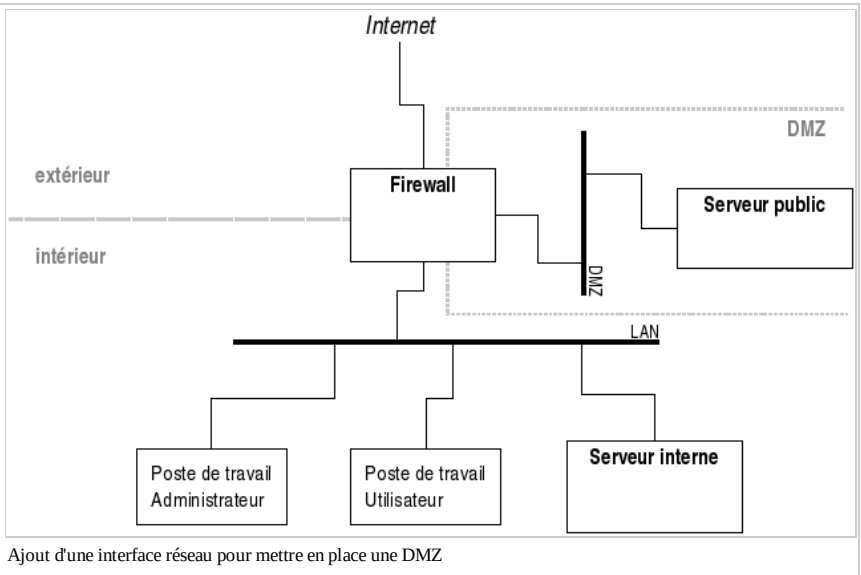
La zone intermédiaire située entre les deux firewall a été baptisée « zone démilitarisée » ou DMZ (demilitarized zone), probablement par analogie avec les zones de terrain situées entre deux frontières. Cette dénomination est restée en usage, même si la mise en œuvre technique a largement évolué.



Niveaux de sécurité multiples

En effet, l'utilisation de deux matériels distincts dans les premières architectures avec DMZ, comme celle présentée figure 8, était tout simplement dûe à la difficulté de disposer de machines disposant de plus deux interfaces réseaux et capables d'exécuter un logiciel firewall. Les capacités techniques des matériels ont rapidement évolué en permettant de s'orienter vers des solutions avec un seul firewall pour la mise en œuvre d'une ou plusieurs DMZ (même si l'utilisation de plusieurs équipements différents a persisté, cette fois-ci pour des raisons de sécurité accrue grâce à la diversification des logiciels). La figure 9 présente une architecture réseau avec une DMZ réalisée en utilisant un seul firewall disposant de 3 interfaces réseau.

Avec une configuration judicieuse de la politique de sécurité du firewall, l'architecture de la figure 9 est équivalente à celle de la figure 8. Le nombre maximal d'interfaces utilisables sur un firewall particulier reste donc un paramètre important qui conditionne les architectures envisageables, et notamment le nombre maximal de DMZ (c'est à dire le nombre maximal de zones de sécurité distinctes utilisables).



Ajout d'une interface réseau pour mettre en place une DMZ

Dans la pratique, ce nombre peut varier couramment entre 2 et 10 environ ; mais ce nombre est parfois beaucoup plus contraint si on prend en considération la vitesse des interfaces (plus de 2 interfaces Gigabit est encore assez rare), le coût des licences logicielles (qui peuvent dépendre du nombre d'interfaces), ou encore la configuration matérielle des machines (la disponibilité ou non de cartes réseaux multi-adapteurs - cartes quad ou bi - peut changer radicalement ce paramètre).

Utilisations pratiques des DMZ

Diversification et haute-disponibilité

Translation d'adresses

Une autre fonctionnalité particulièrement importante des firewall est la translation d'adresses (ou NAT pour Network Address Translation) : c'est la capacité fréquemment associée à ces équipements de transformer les adresses des paquets qui transitent à travers eux, en multiplexant si besoin plusieurs adresses internes

sur un ensemble plus réduit d'adresses externes.

Dans le cas d'une translation d'adresses simple impliquant uniquement les adresses IP source des paquets d'une communication, il est ainsi possible grâce au firewall de substituer à une adresse ip du réseau interne une autre adresse ip' appartenant à la plage d'adresses publiques officiellement affectée à l'organisation concernée, suivant le modèle suivant : *formule à retranscrire*.

Dans le cas où le réseau interne utilise (à juste titre) une plage d'adresse située dans un réseau non-routable (192.168.1.0/24 ou 10.0.0.0/8 par exemple [RFC 1918]), ceci est indispensable pour permettre aux machines du réseau interne de communiquer avec des machines situées sur Internet. Par ailleurs, la substitution étant dynamique, n'importe laquelle des *N* machines du réseau interne peut utiliser momentanément une adresse publique de manière transparente, le nombre maximal de machine pouvant communiquer simultanément avec l'extérieur étant toutefois limité par la taille de l'espace d'adresses public affecté à la translation d'adresses (*P* ci-dessus). Un avantage de cette translation basée seulement sur les adresses IP tient au fait que, pendant la période de communication durant laquelle l'adresse publique est affectée à une machine du réseau interne, celle-ci peut également être contactée depuis l'extérieur (si le firewall l'autorise). Toutefois, dans la pratique, ce mode de translation est assez peu utilisé pour les postes de travail du réseau interne, au profit de celui présenté ci-après. Par contre, c'est ce type de translation qui est couramment utilisé pour les serveurs accessibles depuis Internet, mais dans ce cas de manière statique, avec une affectation permanente de l'adresse IP publique à l'adresse IP interne du serveur, généralement en DMZ.

Les adresses IP source des paquets ne sont pas les seuls éléments utilisables pour réaliser une translation d'adresse : il est également possible de jouer sur les numéros de port source TCP ou UDP utilisés dans la mise en œuvre de communications avec ces protocoles. Dans ce cas, la translation d'adresses suit (pour TCP) le schéma suivant, où l'on voit que l'adresse source TCP/IP complète est substituée : *formule à retranscrire*

Ce multiplexage est surtout naturel vis à vis d'un protocole orienté connexion comme TCP (en se basant sur le port source). Il est également possible sur UDP, dans le cas des protocoles impliquant requête puis réponse (notamment le DNS). Il peut aussi être introduit pour ICMP (surtout pour le ping).

L'intérêt majeur de cette translation, parfois appelée PAT (pour Port Address Translation) est d'offrir des possibilités de multiplexage bien plus larges que dans le cas précédent. En utilisant une seule adresse IP publique (généralement celle de l'interface externe du firewall d'ailleurs) et en jouant sur la large plage de numéro de port source disponible (en théorie, de 1025 jusqu'à 65535 pour TCP ou UDP), il est parfaitement possible de permettre à plusieurs milliers de machines du réseau interne d'accéder simultanément à des serveurs sur Internet par TCP ou UDP. C'est un besoin désormais fréquent compte-tenu de la taille (relativement) réduite de l'espace d'adressage offert par IPv4 (32 bits). Par contre, du fait du mode de translation utilisé, ces machines internes doivent être elles-mêmes à l'origine de la demande de connexion et ne peuvent pas être contactées directement par leurs interlocuteurs depuis Internet, ceux-ci ne pouvant identifier que le firewall.

Ce type de translation d'adresses courant pose parfois des problèmes pour le fonctionnement de certaines applications. L'exemple type est FTP. Dans un échange FTP usuel, si la connexion de contrôle est bien établie à l'initiative du client, les connexions secondaires utilisées pour les transferts de fichiers peuvent être établies à l'initiative du client (mode passif) ou bien du serveur (mode actif), ce dernier mode étant fréquemment le mode par défaut. Dans le cas de la mise en œuvre d'une translation d'adresses, le mode FTP actif ne peut pas fonctionner sans une inspection plus détaillée du déroulement de la connexion par le firewall, ou l'utilisation d'un proxy transparent sur le firewall (ce qui revient un peu au même) pour effectuer un suivi de l'état de la connexion FTP et réagir correctement aux connexions TCP secondaires entrantes. Dans le cas de protocoles courants comme FTP ces fonctionnalités additionnelles sont généralement facilement disponibles, mais cet exemple illustre le fait que la présence du firewall et des fonctions de translation d'adresses ne sont pas anodines ; et les réactions des utilisateurs à ce type de situation sont difficiles à gérer.

Firewall : fonctionnement interne

Par rapport au fonctionnement interne d'un firewall, il est utile de préciser un certain nombre d'éléments qui participent au fonctionnement du logiciel et permettent de mieux comprendre et de mieux administrer ces équipements :

- Tables : chaque firewall capable de réaliser un suivi d'état des connexions gère un certain nombre de tables internes, souvent accessibles à l'administrateur, qui reflètent, à un instant *t*, les différentes connexions connues gérées par l'équipement. On rencontre notamment :
  - Les tables d'état : pour chaque connexion TCP ou pseudo-connexion UDP autorisée, ces tables identifient l'état de la connexion (en cours d'établissement, établie, interrompue, etc.) et permettent : d'abord d'autoriser les paquets nécessaires, mais également de contrôler, voire d'imposer un déroulement « conforme ».
  - Les tables de translation : qui maintiennent la correspondance entre les adresses privées des machines du réseau interne initiant des connexions vers l'extérieur et les adresses ou les numéros de ports choisis par le firewall pour transformer les paquets avant de les acheminer.
- Traces : chaque firewall offre des fonctions de surveillance des flux réseau qui le traversent, fréquemment en liaison avec les règles d'autorisation de sa configuration de filtrage qui permettent de tracer (logging) ou non un paquet autorisé ou rejeté. Ces flux peuvent être très volumineux, notamment dans les cas où l'on souhaite conserver le contenu des paquets réseaux ainsi repérés et générer une charge de traitement importante pour le firewall et les systèmes d'administration associés. Les modes de sélection et la voie d'accès, de transport et de traitement des traces sont des points assez important dans le fonctionnement interne du firewall. Les traces sont pourtant très utiles pour confirmer et identifier des attaques et leurs auteurs.
- Fonctions de normalisation des paquets : outre l'application des règles de filtrage de la politique de sécurité réseau, les firewall réalisent en général également des traitements de normalisation visant à maintenir des flux réseau normaux dans le trafic qu'ils font transiter. Ceci peut parfois impliquer des règles de normalisation un peu brutales (comme le rejet des paquets fragmentés par exemple) dont il est utile de connaître l'existence. Dans la pratique, pour les logiciels les plus répandus, ces fonctions améliorent assez notablement la qualité du flux réseau en isolant en général du trafic réellement anormal.
- Analyses et fonctions avancées : enfin la plupart des firewall vont désormais au-delà de la mise en œuvre du suivi d'état et de la translation d'adresses en permettant parfois de réaliser des fonctions sophistiqués (et plus ou moins utiles) dont nous mentionnons certaines ici.
  - Substitution des numéros de séquence : afin de limiter la prévisibilité des numéros de séquence TCP utilisés par certains systèmes d'exploitation, certains firewall sont en mesure de substituer ceux-ci sur les connexions qu'ils acheminent, ce qui est un exercice assez délicat.
  - Inspection protocolaire : pour répondre aux besoins spécifiques correspondant à des applications et des protocoles largement répandus (comme FTP bien entendu, mais aussi le DNS, H.323, etc.), un certain nombre de firewall sont en mesure d'aller au-delà de l'analyse des en-têtes IP et TCP/UDP des paquets pour examiner également les protocoles de plus haut niveau. Cette inspection protocolaire permet alors de mieux contrôler les flux et de simplifier la configuration du firewall en autorisant tous les flux nécessaires aux communications d'une application.
  - Redirection : une autre manière de répondre efficacement aux besoins des applications de communication complexes, c'est de permettre à des proxy applicatifs présents sur le firewall d'intervenir également sur la communication, et de manière transparente. Dans ce cas, le firewall doit pouvoir faire une redirection des flux autorisés vers des relais applicatifs, et permettre à ces relais de continuer la communication. L'objectif de ce mode de fonctionnement, relativement équivalent au précédent, est alors notamment d'éviter de polluer le logiciel de filtrage lui-même (habituellement étroitement associé aux couches réseau du noyau) avec des fonctions d'analyse plus avancées nécessitant des logiciels de type applicatif.
  - OS fingerprinting : certains travaux récents, notamment sur le firewall pf d'OpenBSD, ont introduit des éléments nouveaux dans les possibilités offertes par les firewall. En couplant la mise en œuvre des règles de filtrage avec des fonctions d'analyse réseau capables de reconnaître la signature de certains types de système d'exploitation, les dernières versions d'OpenBSD sont en mesure d'autoriser ou de limiter un flux de communication en fonction de certaines caractéristiques des machines impliquées dans la communication (et notamment le type de système d'exploitation). Ceci offre des possibilités assez nouvelles, notamment pour lutter contre les systèmes vieillissants.
  - Intégration avec la QoS : étant amené à suivre de la manière la plus précise possible les flux de communication qu'il achemine (au point parfois de nécessiter

une analyse partielle des commandes utilisateurs inscrites dans la communication), le firewall est également très bien placé pour appliquer des règles de qualité de service réseau aux flux qu'il a identifié. Il est donc techniquement souhaitable d'associer des règles de QoS réseau aux règles de filtrage pour répartir la bande passante et accorder des priorités adéquates aux différents flux de communication. Toutefois, cette pertinence technique entre probablement en conflit avec les objectifs commerciaux de nombreux constructeurs (qui associent la sécurité et la qualité de service à des gammes de matériels et des offres différentes) et l'organisation courante de l'administration informatique (qui établit des frontières assez hermétiques et souvent une certaine concurrence entre l'administration de la sécurité et l'administration du réseau). Dans la pratique, les solutions techniques offrant une réelle intégration de la QoS et du firewall sont alors beaucoup plus homogènes dans le domaine du logiciel libre, avec Netfilter pour Linux, et pf avec ALTQ sous OpenBSD ; avec une exception notable : le module FloodGate-1 de CheckPoint (qui mériterait sans doute un intérêt supérieur à celui qu'il attire en général).

Sécurité informatique/Protection réseau et firewall/Aspects architecturaux/Utilisations pratiques des DMZ

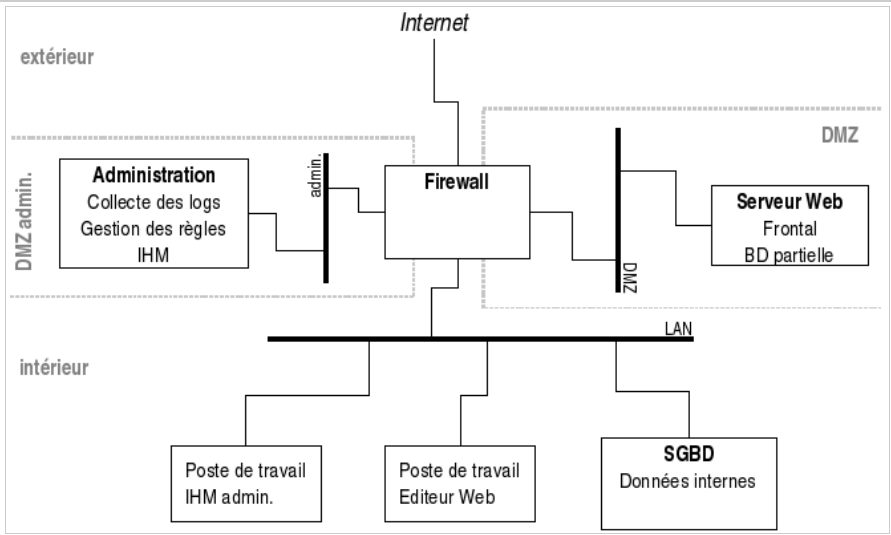
Dans cette section, nous présentons plusieurs utilisations possibles pour les DMZ en ajoutant progressivement à l'architecture de la figure 9 différents types de DMZ pour aboutir à une architecture complexe, mais assez représentative d'un point d'accès réel pour le système d'information d'une grande entreprise.

Administration

Le premier usage qui peut être fait pour une DMZ consiste à isoler dans un sous-réseau protégé les différents moyens d'administration du firewall lui-même. On peut ainsi positionner dans une DMZ dite « d'administration » certaines machines utilisées, par exemple, pour paramétrer les règles du (ou des) firewall du système d'information, pour stocker durablement les traces collectées (paquets rejetés par exemple), pour mettre à jour le logiciel du firewall ou pour mettre en œuvre des procédures d'authentification associées aux équipements de sécurité eux-mêmes.

Dans la pratique, même si les postes de travail des administrateurs, situés sur le réseau local interne, sont également impliqués dans l'administration du firewall, notamment pour l'affichage d'une console de gestion, une DMZ d'administration est fréquemment nécessaire pour l'un ou l'autre des fonctions mentionnées précédemment. De plus, quand plusieurs firewall existent dans le système d'information, une seule DMZ d'administration peut permettre d'héberger les services d'administration nécessaires pour ces différents équipements. Le paramétrage de chaque firewall doit alors prendre en compte le besoin de protéger spécifiquement le trafic vers ce sous-réseau tout au long de son trajet.

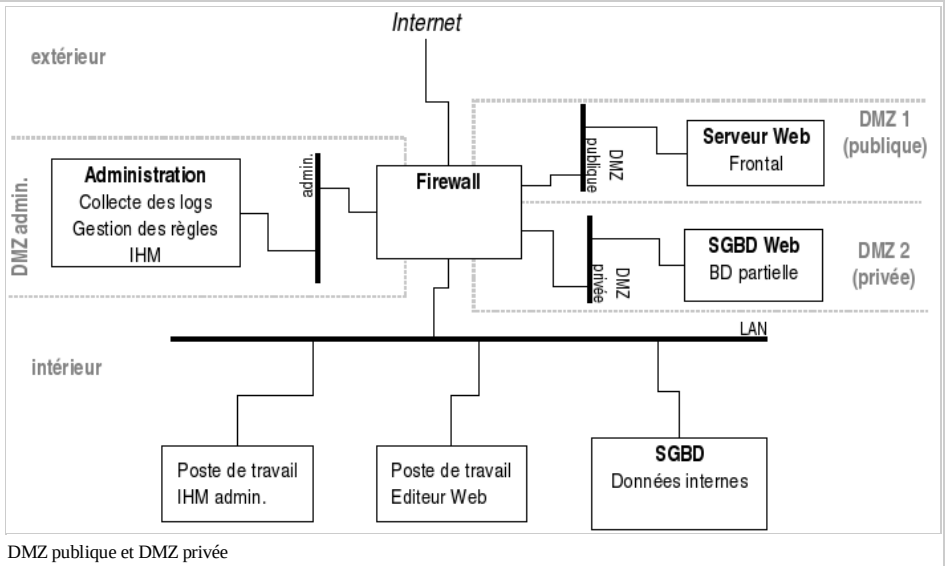
En tout état de cause, quand elle existe, cette DMZ d'administration doit être prévue pour se situer à un niveau de sécurité très élevé, très certainement le plus élevé de toutes les zones du réseau (à part peut-être le système d'exploitation des firewall eux-mêmes). L'accès doit y être très contrôlé au niveau réseau, et le périmètre physique du sous-réseau concerné doit pouvoir être facilement identifié (ce qui proscrit, par exemple, les accès RTC ou via un réseau sans fil directement sur cette DMZ).



Utilisation d'une DMZ pour l'administration sécurité

Protection à plusieurs niveaux

Dans le cas d'un service réseau ouvert sur Internet, comme un serveur Web par exemple, l'utilisation de DMZ peut également permettre d'offrir une protection à plusieurs niveaux pour ce service. Le serveur HTTP lui-même doit, bien sûr, apparaître dans une DMZ accessible depuis Internet.



DMZ publique et DMZ privée

Par contre, les services secondaires utilisés par ce serveur HTTP pour fournir le service Web, comme une base de données par exemple, peuvent être isolés dans une deuxième DMZ, distincte. Ainsi, on peut distinguer un sous-réseau accessibles par des flux en provenance d'Internet (une « DMZ publique ») et un autre

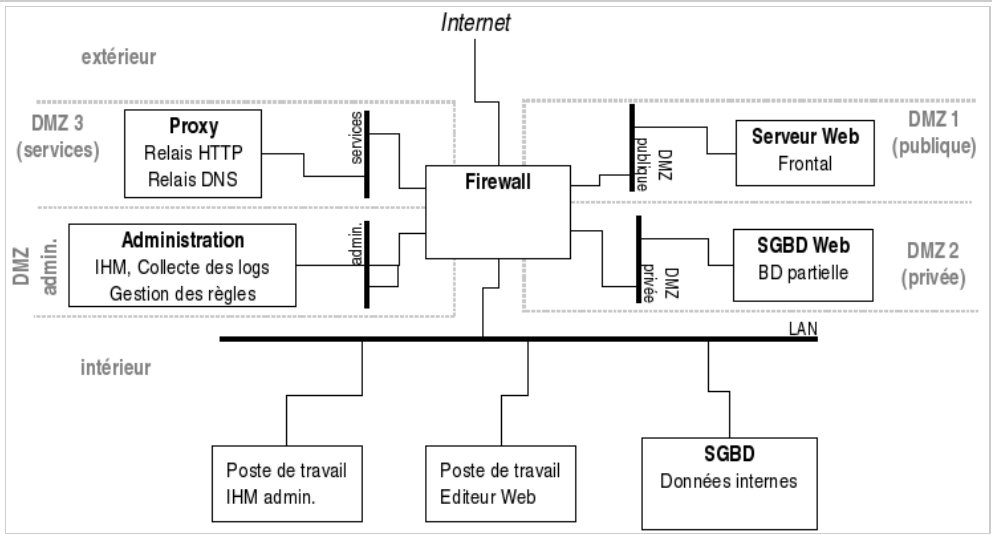
sous-réseau accessible seulement depuis la DMZ publique constituant une « DMZ privée ». Seules les machines devant être directement accessibles depuis Internet sont situées sur la DMZ publique, les serveurs de base de données (ou éventuellement des serveurs d'applications) utilisées par ces machines étant eux placés dans une DMZ distincte.

Cette séparation par niveau de sensibilité des différentes machines éventuellement impliquées dans la mise en oeuvre d'un service public est à distinguer d'un cloisonnement entre services différents également rendu possible par l'utilisation de différentes DMZ. En effet, par exemple, si deux serveurs HTTP sont installés dans le système, et placés dans des DMZ différentes, on envisage en fait une architecture différente, comptant deux DMZ publiques. Bien évidemment, on peut utiliser les deux approches simultanément et disposer de plusieurs DMZ publiques et plusieurs DMZ privées associées. C'est l'évaluation des risques et des vulnérabilités éventuelles des différentes applications et des différents systèmes mis en jeu qui permet de choisir une bonne architecture. (Le nombre maximal d'interfaces du firewall, et donc de DMZ, est aussi une contrainte importante.)

Relais

Les DMZ de type publique ou privée sont généralement associées à des flux réseaux entrants du point de vue du système d'information, c'est à dire des accès en provenance d'Internet vers certains serveurs publics.

Des DMZ spécifiques peuvent également être utilisées pour faire transiter des flux sortants du réseau local vers Internet. Dans la majeure partie des cas courants, une seule de ces DMZ est envisagée. Nous désignerons ici ce type de DMZ sous le nom de « DMZ (de) service ».



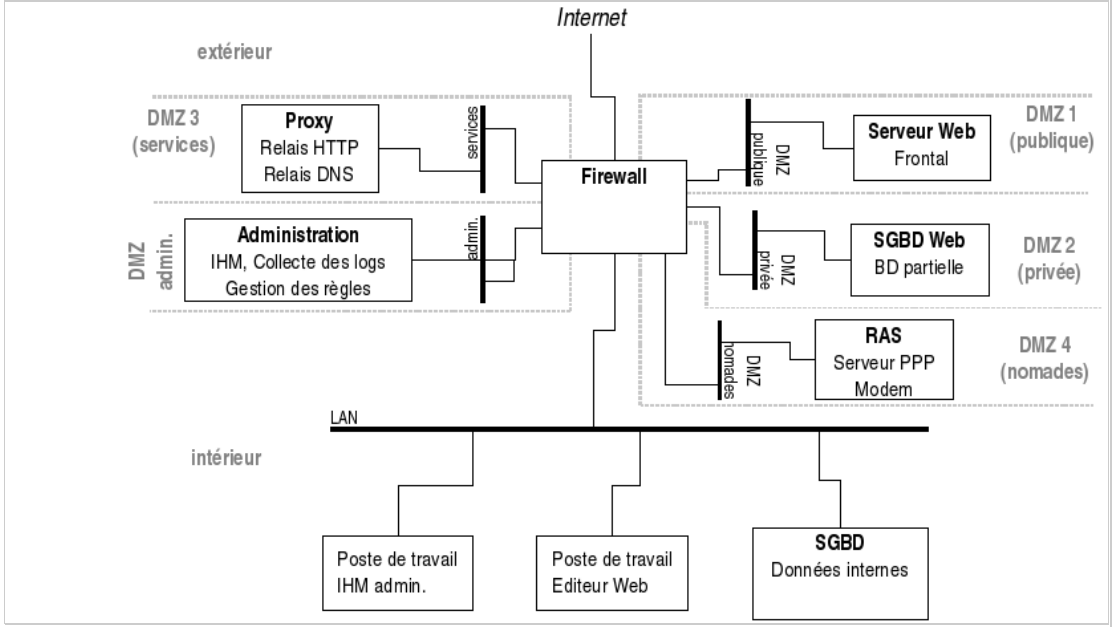
Utilisation d'une DMZ pour la mise en place de relais

Une première application évidente d'une DMZ de service, présentée dans le figure ci-dessus, consiste à installer dans un sous-réseau protégé des serveurs relais DNS ou relais HTTP (proxy). Ainsi isolés, ces serveurs sont protégés d'éventuels abus provenant du réseau interne. Il est également possible de mieux maîtriser leur fonctionnement et leurs interactions avec le réseau interne.

Accès externes

Dans la plupart des organisations, il est courant de vouloir également offrir un service d'accès au réseau interne pour des utilisateurs « nomades », généralement équipés d'ordinateurs portables fournis par l'organisation. Il s'agit de répondre aux besoins d'utilisateurs itinérants ou de l'encadrement. Pour l'instant, la plupart du temps, ces accès s'effectuent via une liaison téléphonique RTC, soit aboutissant directement à des modems de l'organisation, soit à un service d'accès fourni par un opérateur.

Comme indiqué dans la figure suivante, il est intéressant d'isoler le point d'arrivée de ces accès entrants dans une DMZ spécifique, que nous appellerons ici une « DMZ nomades ». Ceci permet de contrôler plus précisément les différentes ressources du réseau interne utilisables par ces accès nomades, par exemple en filtrant les accès à destination des principales ressources utilisées (principalement le serveur de messagerie, certains serveurs de fichiers) et en limitant les protocoles réseau utilisables. Cette DMZ peut également contenir les systèmes d'authentification spécifiques à ces accès nomades. En effet, il est généralement souhaité de renforcer l'authentification des utilisateurs nomades étant donné les risques associés à ce type d'accès qui permet d'entrer au cœur même du réseau interne.



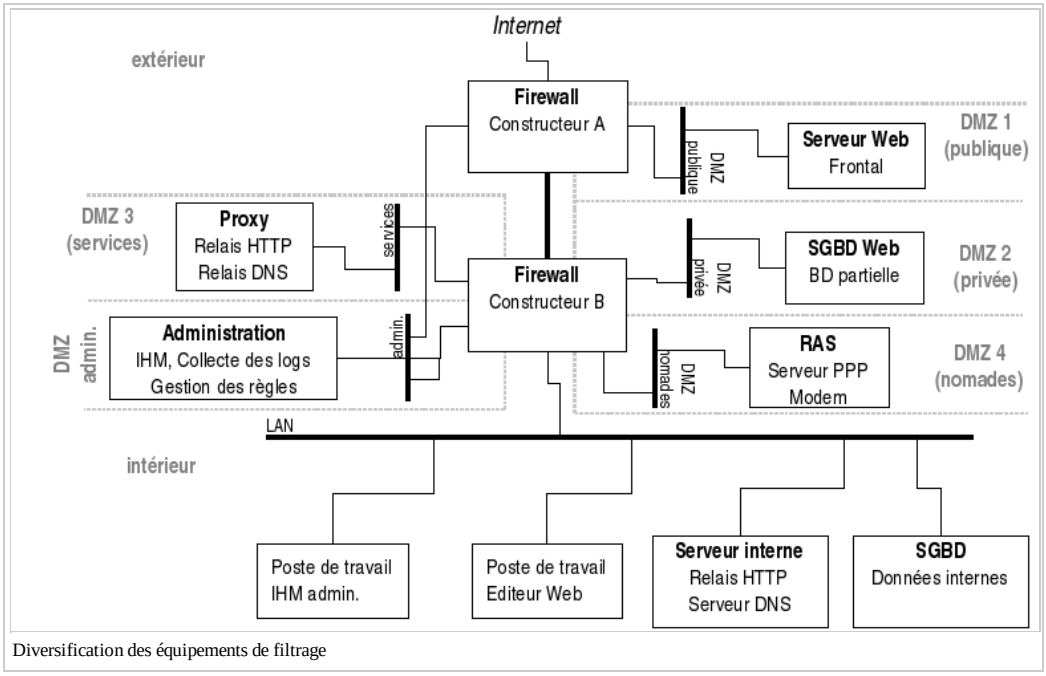
Utilisation d'une DMZ pour contrôler les accés distants

Sécurité informatique/Protection réseau et firewall/Aspects architecturaux/Diversification et haute-disponibilité

Quand l'architecture réseau associée au firewall atteint le degré de complexité présenté précédemment, son rôle dans le système informatique de l'organisation associée commence à devenir assez critique. Cette sensibilité peut être compliquée par des exigences de sécurité très élevées, ou plus fréquemment par un besoin de forte disponibilité du firewall dont la défaillance peut entraîner la coupure de la plupart services réseaux en contact avec l'extérieur. Dans ces deux cas, il est possible de répondre aux besoins en combinant l'utilisation de plusieurs firewall.

Diversification

Un besoin de sécurité très élevé peut parfois être rempli en positionnant deux firewall de constructeurs différents en cascade l'un après l'autre, comme indiqué sur la figure suivante.



Diversification des équipements de filtrage

La logique sous-jacente à cette architecture s'appuie sur la diversification des logiciels et éventuellement des plate-formes : si une vulnérabilité ou une faille de sécurité grave est révélée sur l'un des équipements, l'autre équipement est en mesure de pallier à ce problème de sécurité.

Le positionnement des DMZ autour des deux équipements n'est pas toujours évident. En toute logique, chaque flux réseau devrait être contraint à traverser les deux firewall l'un après l'autre pour profiter de la sécurité accrue offerte par la diversification. Toutefois, la configuration des équipements devient alors généralement très complexe (ce qui est également une source d'erreurs de configuration, et donc parfois de problèmes de sécurité). On positionne donc parfois de manière plus naturelle les différentes DMZ, en fonction du niveau de sécurité souhaité pour les systèmes informatiques qu'elles contiennent. (Ceci permet également parfois d'augmenter le nombre global d'interfaces disponibles pour mettre en place des DMZ.) Par exemple, dans le figure 14, la DMZ publique contenant les serveurs HTTP directement accédés depuis Internet reste protégée par le seul firewall externe ; et les flux des portables itinérants aboutissant dans le DMZ nomades ne traversent que le firewall interne pour entrer sur le réseau local. Par contre, les flux sortants vers Internet, ou les échanges entre les serveurs HTTP publics et la base de données située dans le DMZ privée doivent transiter au travers des deux équipements. Bien évidemment, on peut faire varier ce positionnement. Par contre, en règle générale, il est souhaitable de connecter directement la DMZ d'administration aux deux firewall pour faciliter leur gestion.

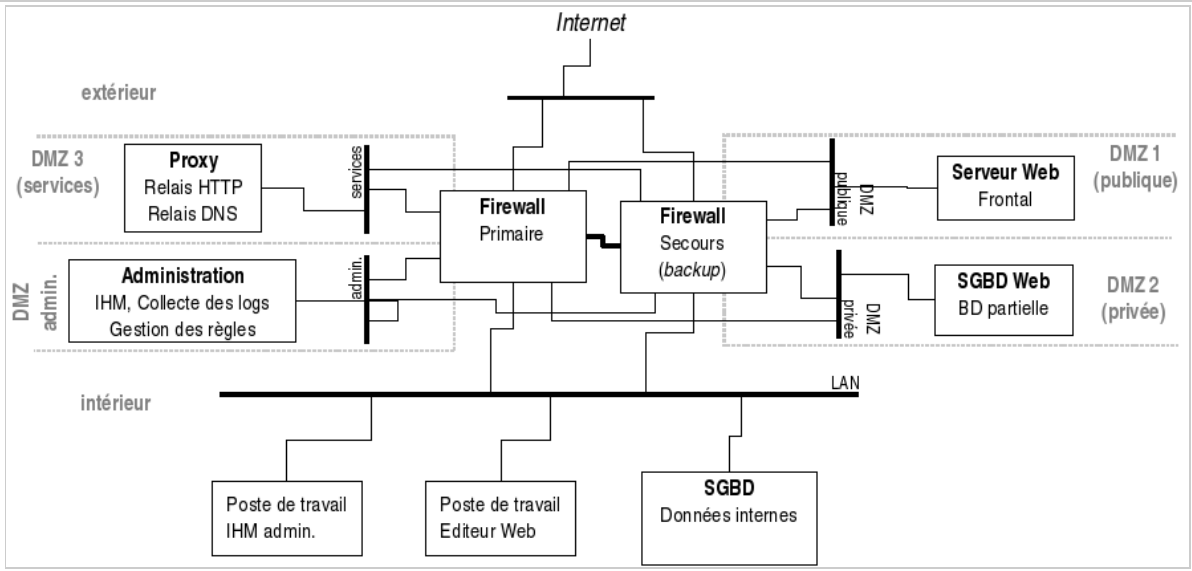
Dans la pratique, la difficulté de l'administration de ce type d'architecture conduit généralement les équipes d'exploitation à privilégier l'un des deux équipements (souvent le plus facile à administrer). Celui-ci met alors en œuvre l'essentiel du filtrage fin des flux réseaux entre les différentes DMZ. Le deuxième firewall (généralement le firewall interne) est doté d'une configuration plus générale et beaucoup plus stable (par exemple en diode, comme présenté au 3.1.3.1). Il joue alors le rôle d'une deuxième ligne de protection, assurant notamment la protection du réseau interne dans le cas où l'équipement principal présente une grave faille de sécurité.

Bien que l'utilisation de la diversification puisse parfois sembler un peu excessive, c'est une approche que l'on rencontre finalement assez fréquemment.

Redondance

Un besoin de forte disponibilité pour les services réseaux offerts par l'architecture de sécurité, et notamment les services Web accessibles depuis Internet peut également conduire à la mise en place d'architectures dites de « haute disponibilité » impliquant deux équipements firewall. Toutefois, dans ce cas, il s'agit d'équipements de même type, et généralement dotés d'un logiciel et de connexions réseau spécifiques, dédiées à la gestion de la redondance.

Pour répondre au besoin de disponibilité, les deux firewall fonctionnent généralement en mode de redondance passive (ou secours chaud, ou primary/backup). A un instant donné, l'un des deux équipements assure les fonctions de filtrage tandis que l'autre se tient prêt à prendre le relais en cas de défaillance du premier. Pour cela, l'équipement de secours doit disposer d'une version à jour de la liste des règles de filtrage, ainsi qu'une version des tables d'états gérées dynamiquement par le firewall pour le suivi des connexions en cours. Les deux équipements doivent également disposer d'un moyen de se surveiller mutuellement.



Mise en redondance de deux firewall

Ces différents besoins sont généralement remplis par la mise en place d'une connexion réseau directe entre les deux firewall, identifiée en gras sur la figure précédente.

Bien évidemment, toutes les connexions réseaux reliant les DMZ doivent également être doublées de manière à permettre à chacun des deux firewall d'assurer le filtrage ; et il importe aussi de prendre garde à ne pas compromettre les gains de disponibilité permis par la redondance des firewall en introduisant d'autres points de défaillance unique. (Par exemple en plaçant un concentrateur sur le lien direct entre les deux firewall, ou en les connectant tous les deux à la même source d'alimentation électrique, etc.)

L'architecture réseau s'en trouve assez compliquée, mais les gains sont intéressants. Outre l'accroissement de la protection du filtrage de sécurité par rapport à des défaillances accidentelles, la redondance peut également permettre la réalisation de certaines opérations d'administration en deux étapes successives sans interruption du service : par exemple, la mise à jour du logiciel des firewall.

Enfin, dans certains cas, on peut opter pour des solutions de redondance active (ou « partage de charge ») consistant à faire fonctionner simultanément les deux firewall en répartissant les flux réseaux entre eux. Dans ce cas, les capacités maximales de traitement de l'architecture de sécurité peuvent être accrues (hors défaillance bien entendu, ce qui est un point généralement négligé) et éventuellement étendues en ajoutant d'autres éléments. Toutefois, il nous semble que ces solutions s'écartent parfois du besoin originel en confondant quelque peu les questions de performance (qui sont généralement mieux réglées autrement qu'en multipliant les machines) et de disponibilité.

Sécurité informatique/Systèmes d'authentification

Nous allons à présent nous intéresser aux mécanismes d'authentification utilisables dans les systèmes informatiques. De manière générale, on peut recenser différentes catégories d'authentification :

- Codes d'accès (« Sésame, ouvre-toi ! ») : nous ferons entrer dans cette catégorie tous les modes d'accès qui n'offrent quasiment aucune sécurité mais qui s'appuient néanmoins sur la connaissance d'un mot de passe particulier (dans la pratique, celui-ci peut être aussi varié qu'une adresse IP, un mot du dictionnaire, un nom de compte ésotérique, etc.). Dans la pratique, ces codes d'accès sont finalement très répandus. A moins qu'ils ne soient *à usage unique*, ils n'offrent pas plus qu'une illusion d'authentification, et nous ne les aborderons pas plus avant.
- Nom d'utilisateur / mot de passe : la technique d'authentification qui reste la plus courante est celle consistant à définir un nom d'utilisateur et un mot de passe associé. Le premier permet d'identifier l'utilisateur et est fourni de manière déclarative. Le second est ensuite demandé par le système pour valider l'identification déclarée en comparant une information supposée connue seulement de l'utilisateur légitime avec une information stockée sur le système. En toute rigueur, le mot de passe ne doit pas être connue en clair du système lui-même et surtout de ses administrateurs. Afin de limiter l'impact d'un vol et l'omniscience des administrateurs système, les techniques de cryptographie permettent de se limiter au stockage d'un hash du mot de passe sur le système.
- Clef publiques/clefs privés: : les technique d'authentification basées sur l'utilisation des possibilités offertes par la cryptographie asymétrique, associant une clef publique connue de tous et une clef privée détenue seulement par l'utilisateur (ou un objet dont il dispose) sont celles fournissant à l'heure actuelle, dans la pratique, les moyens d'authentification les plus résistants du point de vue technique. Des exemples courants dans ce domaine sont l'utilisation de RSA, DSA pour des applications comme SSH (authentification des machines et des utilisateurs), IKE (établissement de tunnels chiffrés IPSEC), l'authentification par carte à puce sur les systèmes d'exploitation, etc.
- Authentification « forte » des utilisateurs : les techniques d'authentification dites « fortes » combinent l'utilisation de protocoles d'authentification spécialement étudiés (faisant appel à des algorithmes cryptographiques sérieux) et des éléments logiciels ou matériels permettant aux utilisateurs de mettre en œuvre assez facilement ces protocoles :
  - mots de passe jetables intégrant une composante horaire ;
  - mots de passe jetables (type S/Key, par logiciel ou calculette) ;
  - cartes à puce et token USB.
- Certaines techniques d'authentification abusent en fait de l'utilisation d'un dispositif matériel plus ou moins sophistiqué détenu par l'utilisateur pour ressembler à la famille précédente : pistes magnétiques ISO, numéros d'identification RFID (cartes sans contact), etc. ; mais elles ne sont pas si « fortes » que cela.

On notera enfin que ces différentes techniques d'authentification sont largement orientées vers l'authentification de l'utilisateur auprès du système, voire d'un système auprès d'un autre système. L'authentification du logiciel s'exécutant à un instant donné auprès de l'utilisateur (notamment au moment de l'authentification de l'utilisateur), ou même la vérification de l'intégrité du logiciel au moment de l'installation sont encore largement ignorés dans la pratique. Au moins, ceci confirme l'inusable actualité de la tactique du cheval de Troie, pourtant millénaire.

Hardware tokens

Crypto hardware tokens

Les techniques d'authentification utilisant des dispositifs matériels spécifiques intégrant des fonctions de calcul cryptographiques (notamment de cryptographie asymétrique) sont une classe importante de moyens d'authentification pour les utilisateurs.

De manière générale, ces dispositifs sont souvent désignés sous le nom de « carte à puce », bien que cette dénomination commence à devenir trompeuse, toutes les puces n'intégrant pas nécessairement de fonctions cryptographiques, a fortiori de cryptographie asymétrique, et certaines de ces puces étant désormais proposées sur d'autres supports que des cartes.

Dans la pratique, une carte à puce est détenue par l'utilisateur qui partage avec elle un code d'identification (souvent appelé PIN), généralement numérique ou alphanumérique. Ce PIN permet à l'utilisateur de déverrouiller la carte pour une session d'authentification spécifique et protège celle-ci (de manière limitée) en cas de vol. La carte à puce elle-même réalise une authentification avec la machine hôte via un protocole d'authentification du type de ceux utilisant la cryptographie RSA ou DSA - c'est à dire en utilisant un algorithme cryptographique asymétrique permettant à la carte à puce de prouver son identité sans révéler la clef privée qu'elle détient. Suivant les modèles, cette clef privée est stockée dans une zone mémoire protégée de la puce[1] et est même générée sur la puce elle-même. Dans ce dernier cas, à aucun moment la clef privée ne sort de la carte à puce et constitue donc un secret de très bonne qualité.

Les standards industriels semblent également converger vers un stockage du bi-clef d'authentification sous le format des certificats X.509. Enfin, les cartes à puce nécessitent un dispositif de lecture spécifique, un lecteur de cartes, installé sur la machine hôte. Pour pallier au besoin de cet équipement additionnel, le format du dispositif à évolué pour tirer partie des ports USB disponibles depuis quelques années sur la plupart des micro-ordinateurs, et on rencontre désormais ce dispositif sous la forme d'une clef USB. Par ailleurs, ces clefs USB d'authentification présentent les mêmes caractéristiques techniques qu'une carte à puce, quand elles utilisent le même type de microprocesseur. Mais ce format USB peut également être utilisé par d'autres types de « puces » (les clefs mémoires simples par exemple, sont désormais très répandues). Des dispositifs matériels intégrant des fonctions cryptographiques sont également disponibles pour réaliser l'authentification utilisant des mots de passe jetables (dont l'intérêt est de pouvoir fonctionner avec un protocole d'authentification non-sécurisé pré-existant). On rencontre parfois des dispositifs de ce type, souvent appelés « calculettes ». Ces calculettes nécessitent généralement un PIN pour les déverrouiller. Des versions logicielles de ce type de dispositif peuvent également être utilisées, si une confiance suffisante dans la machine sur laquelle est exécutée ce logiciel est acquise.

Autres badges d'identification

D'autres types de dispositifs matériels d'authentification sont également relativement courants, bien qu'ils ne soient pas toujours utilisés pour l'authentification sur un système d'exploitation, mais plutôt pour le contrôle d'accès physique ou d'autres fonctions courantes (pointage horaire, paiement cantine, suivi de production, etc.). Le plus courant est probablement la carte à piste magnétique ISO (souvent appelée « badge »), mais d'autres dispositifs sont également assez répandus (badges Weygand, inductifs, etc.). La limite entre ces « badges d'identification » et la « carte à puce d'authentification » n'est parfois pas facile à reconnaître.

Dispositifs « sans contact » (RFID, etc.)

Zoom sur les tokens à puces cryptographiques

Mots de passe et attaque des mots de passe

Malgré l'existence de dispositif matériels d'authentification, l'authentification par mot de passe reste la technique la plus répandue dans les systèmes informatiques. Sa prolifération a même donné lieu à la création d'un marché pour des produits de gestion des différents mots de passe dont un utilisateur peut être détenteur, et qu'il n'arrive plus à mémoriser seul. Cette technique sépare l'identifiant : le nom d'utilisateur fourni de manière déclarative ; et l'authentifiant : un mot de passe secret.

Stockage

Cet authentifiant est stocké à disposition du système d'authentification, qu'il s'agisse du logiciel permettant d'ouvrir des sessions utilisateur au niveau du système d'exploitation (login) ou bien au niveau d'un service applicatif (par exemple un serveur HTTP) ou d'un SGBD. Il est d'abord important de distinguer différentes formes de stockage du mot de passe sur le système. En effet, certains utilisateurs (les administrateurs par exemple, ou les opérateurs des systèmes de sauvegarde), ou parfois un grand nombre d'utilisateurs (tous les utilisateurs dans les anciennes versions d'Unix[2]) peuvent avoir accès à cette zone de stockage. Malgré tout, un tel accès ne doit pas leur permettre de découvrir facilement le mot de passe des autres utilisateurs, ce qui leur donnerait un moyen d'usurper totalement leur identité.

On peut distinguer :

- un stockage sous forme « obscurcie », utilisant un encodage spécifique, mais réversible : c'est par exemple le cas de certains équipements réseau Cisco qui font figurer les mots de passe sous cette forme dans un état de leur configuration — d'autres constructeurs peuvent certainement utiliser ce type de stockage — qu'il est important de savoir distinguer de celles utilisant un véritable mécanisme cryptographique ;
- un stockage sous forme chiffrée, par exemple en utilisant une fonction cryptographique à sens unique (*secure hash*) comme le DES d'une valeur constante, MD5, SHA-1, etc. ;
- un stockage sous une forme chiffrée résistante, c'est à dire spécifiquement adaptée pour résister à une attaque par dictionnaire, en démultipliant l'espace de recherche possible (c'est à dire en pratique en ajoutant un salt aléatoire de longueur suffisante au mot de passe fourni par l'utilisateur) et en ralentissant le calcul de la fonction cryptographique pour limiter la vitesse des essais réalisables ;
- enfin, il faut réaliser que ces mots de passe sont parfois stockés directement en clair sur le système ; par exemple dans le cas des applications Web, c'est encore une pratique largement répandue de stocker en clair les mots de passe dans la base de données associée, dans ce cas tous ceux ayant accès à ces tables, notamment les développeurs, sont en mesure d'usurper l'identité des utilisateurs.

Caractéristiques

Du point de vue de l'usage terminologique, il importe de distinguer un mot de passe d'un(e) « *passphrase* » ou un PIN. Dans les deux cas, il s'agit bien de l'équivalent d'un mot de passe ; mais l'usage consacre généralement le terme PIN pour désigner un code numérique servant à déverrouiller une carte à puce, et le terme *passphrase* pour désigner le mot de passe permettant de déverrouiller l'accès à une clef privée stockée sur disque (sous forme chiffrée) par exemple dans un anneau de clef OpenPGP ou la partie privée d'un certificat X.509[3]. La deuxième dénomination fait également référence à une méthode de choix du mot de passe (utilisant une phrase comme support mnémonique) permettant d'arriver à des mots de passe de bonne qualité.

La technique d'authentification par mot de passe étant encore très largement répandue, dans un souci pratique, il faut s'intéresser aux caractéristiques d'un bon mot de passe. Selon nous, un bon mot de passe combine les caractéristiques suivantes :

- il est personnel : c'est à dire spécifique à chaque individu et connu de lui seul (il ne peut donc être « prêté ») ;
- il doit être fiable : c'est à dire durablement mémorisé par son détenteur, malgré des périodes parfois longues de non-utilisation ;
- enfin, pour avoir un rôle réel du point de vue de la sécurité, il doit être résistant : c'est à dire qu'il ne doit pas être facile à deviner pour un tiers (qu'il s'agisse d'un humain ou d'une *machine*).

Ces caractéristiques mettent en lumière les principales difficultés que posent la technique d'authentification par mot de passe. Le caractère personnel s'oppose à beaucoup d'usages, notamment en entreprise, où la délégation d'accès est nécessaire (par exemple en cas d'absence) et souvent impossible à réaliser par les fonctions du système. La fiabilité d'un mot de passe, dans un cerveau humain standard, s'oppose assez directement à sa complexité, dont on comprend pourtant qu'elle est nécessaire pour un minimum de résistance. La résistance à des attaques automatiques devient de plus en plus difficile. La puissance croissante des machines permet désormais d'effectuer jusqu'à plusieurs centaines de milliers d'essais par seconde ce qui permet d'obtenir très rapidement tout mot de passe qui n'a pas été choisi avec soin. Enfin, une utilisation judicieuse de la psychologie humaine peut aussi permettre de deviner le mot de passe d'un tiers : en tout état de cause, la manière la plus simple d'obtenir le mot de passe de quelqu'un reste de le lui demander.

Attaque du mot de passe

L'attaque des mots de passe peut donc s'effectuer de multiples manières :

- Les difficultés de mémorisation rencontrées par les utilisateurs les conduisent souvent à noter leur mot de passe, ou plutôt leurs mots de passe. Dans ces conditions, et c'est probablement la technique la plus simple, il faut commencer par chercher les mots de passe dans la poubelle, sous le clavier, dans les agendas, les PDA, voire tout simplement sur les autocollants entourant l'écran.

- Amener un utilisateur à vous confier son mot de passe peut sembler une stratégie simpliste, elle révèle pourtant toute son efficacité quand elle est menée avec des scénarios suffisamment sophistiqués, en utilisant les techniques dites de *social engineering* (notamment par téléphone). On peut par exemple envisager de se faire passer pour un RSSI souhaitant vérifier le bon fonctionnement de son outil d'attaque par dictionnaire (quoiqu'à notre connaissance, cette stratégie-là n'ait jamais marché).

- Parmi les méthodes techniques de vol d'un mot de passe, la première consiste à essayer d'intercepter le mot de passe au moment où celui-ci est entré par l'utilisateur. Ceci peut être envisagé de plusieurs façons :
  - en conduisant l'utilisateur à exécuter un programme malveillant qui pourra tenter de convaincre l'utilisateur de lui donner son mot de passe - c'est à dire utiliser un Cheval de Troie (en utilisant n'importe quelle animation amusante pour la partie visible du programme afin d'inciter l'utilisateur à la négligence) ;
  - en utilisant un enregistreur clavier, qu'il s'agisse réellement d'un dispositif matériel d'espionnage (une caméra, une indiscrétion) ou d'un logiciel déployé par exemple via un (autre) Cheval de Troie.

- Il est parfois possible d'inverser le codage des mots de passe quand la forme stockée sur l'équipement n'est pas spécifiquement protégée (pour une raison ou pour une autre). C'est par exemple le cas pour les mots de passe utilisateur stockés dans les configurations des routeurs ou des switches utilisant l'IOS Cisco[4] et figurant notamment dans leurs sauvegardes. Dans ce cas comme dans le suivant, il est nécessaire d'obtenir en préalable accès à la zone de stockage de ces mots de passe (d'une manière ou d'une autre).

- Enfin, à partir de la base de données système contenant la forme protégée des mots de passe on peut envisager de mener une attaque par dictionnaire (*password cracking*) permettant de découvrir les mots de passe des utilisateurs et probablement de rebondir vers d'autres systèmes (les utilisateurs utilisant en général des mots de passe identiques ou similaires d'un système à un autre). Cette attaque, assez sophistiquée, présente un certain nombre de caractéristiques :
  - Comme nous l'avons dit, elle nécessite en préalable le vol de la forme stockée, chiffrée en général.
  - L'attaque consiste à réaliser des essais successifs d'authentification par rapport à un dictionnaire pré-établi en calculant la forme chiffrée (comme le fait le logiciel d'authentification) et en comparant chacun des mots du dictionnaire avec les différents mots de passe chiffrés qui ont été dérobés.
  - Les logiciels permettant de mettre en œuvre ce type d'attaque permettent généralement d'aller au-delà de l'utilisation d'un dictionnaire simple et permettent d'étudier un espace de recherche plus large en prenant en compte des règles de combinaison simples imitant celles utilisées par les utilisateurs pour

« compliquer » leur mot de passe (mot à l'envers, ajout d'un ou deux chiffres, etc.).
- La puissance des machines aidant et compte tenu d'un haut niveau d'optimisation pour les logiciels d'attaque, la recherche exhaustive est désormais accessible en un temps assez court pour les mots de passe composés de caractères alphanumériques ou des signes de ponctuation courants (avec une longueur minimale de huit en général).
- La technique est surtout intéressante quand elle peut être appliquée à tout un ensemble de comptes utilisateurs et les optimisations techniques des logiciels d'attaque visent à faciliter ces tentatives en parallèle. Par exemple, une optimisation évidente consiste à utiliser les techniques les plus probables en premier de manière à obtenir les mots de passe simples le plus vite possible.
- Bien que d'un principe assez simple, cette attaque est une forme directe d'une attaque cryptographique plus générale (*codebook-based*) consistant à pré-calculer un ensemble de textes chiffrés à partir d'un dictionnaire (ou de séquences aléatoires de longueur fixée) pour créer un « livre de code » lequel est ensuite utilisé afin d'accélérer le déchiffrement d'un message particulier. Un algorithme cryptographique ou une fonction de dispersion solides doivent pouvoir résister à ce type d'attaque, notamment en rendant impossible la construction d'un codebook utile et de taille significativement inférieure à l'ensemble des textes chiffrés possibles. Une attaque de ce type a récemment mis en péril l'ensemble du système d'authentification d'un système d'exploitation très répandu[5] [6]).
- Bien évidemment, dans la pratique, le choix du dictionnaire est important pour une efficacité accrue (prénoms, langue d'origine du dictionnaire, acronymes usuels sont à intégrer). C'est également par rapport à un dictionnaire stable qu'une évaluation régulière du niveau de solidité des mots de passe peut être effectuée ; cette fois-ci par les administrateurs de sécurité.

Prévention et évaluation

Parmi les différentes techniques d'attaque, un certain nombre s'appuient sur la méconnaissance des utilisateurs des vulnérabilités de la technique d'authentification qu'ils utilisent couramment, ainsi qu'une tendance à sous-estimer les abus qui peuvent être effectués en usurpant leur identité. Face à ce type de danger, c'est d'abord par la sensibilisation des utilisateurs et la formation des administrateurs que l'on peut agir ; par exemple en prenant pour point de départ l'exposé d'une « bonne » méthode de choix d'un mot de passe, soin qui met de fait en avant l'importance du secret et de la qualité de l'authentifiant…

Le mode de stockage des mots de passe sur les systèmes où l'authentification est réalisée ou dans le système d'information (sauvegardes) est également un point technique qui est mis en exergue par une étude plus détaillée de la gestion des mots de passe. Il est parfois possible de choisir entre différents modes de fonctionnement et, dans tous les cas, le détail des algorithmes utilisés par les différents systèmes montre de larges disparités qui pourraient être un critère de choix d'une technique d'authentification. C'est aussi un indice sur le niveau général de confiance que l'on peut accorder dans le système d'exploitation concerné.

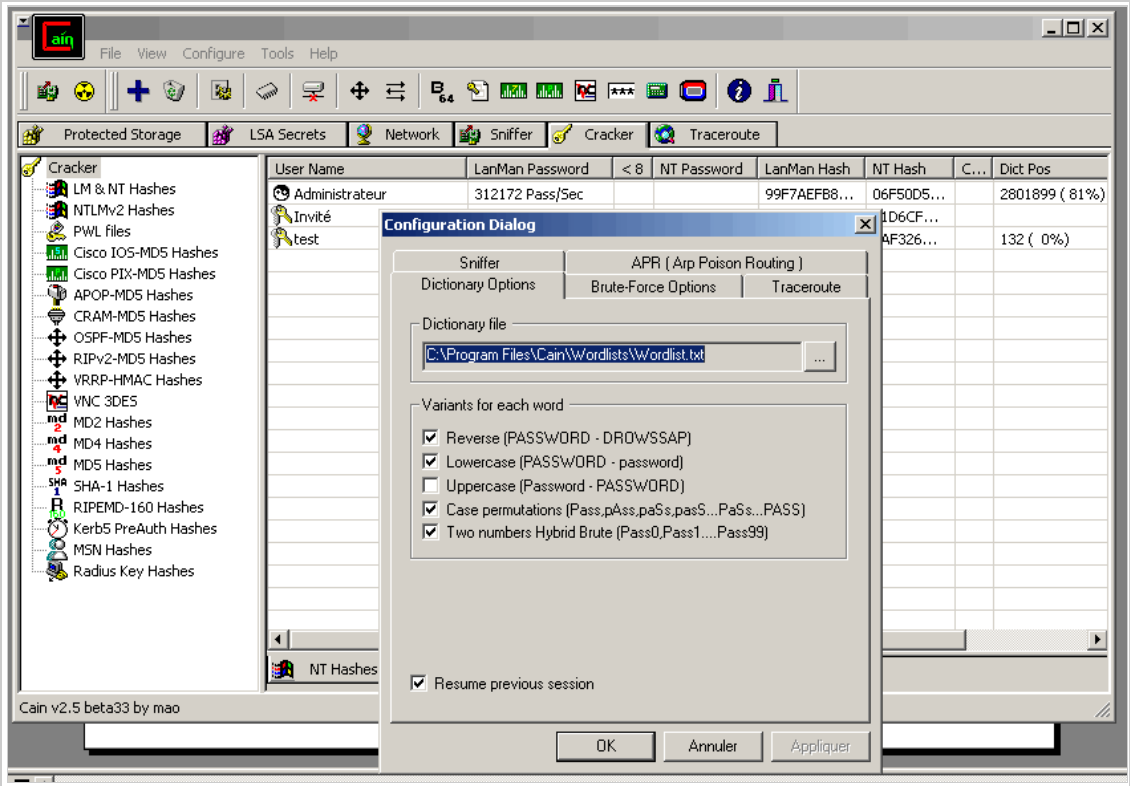
L'examen des algorithmes utilisés est bien évidemment source d'information, mais la mise en œuvre effective de l'attaque par dictionnaire avec un logiciel librement disponible l'est également : elle permet d'offrir un point de vue tout à fait concret sur l'efficacité combinée des choix des utilisateurs et des algorithmes de stockage en usage dans le système d'information. A notre connaissance, ces données offrent également un moyen d'alerte utile au niveau des directions. On pourrait également envisager de les utiliser pour offrir une motivation aux utilisateurs et les encourager à adopter des politiques de choix judicieuses ; mais c'est un terrain délicat sur lequel nous ne nous sommes encore jamais aventurés et qu'il importe certainement d'inscrire dans un « plan de communication » réfléchi pour susciter une réaction positive[7].

Une attaque régulière des mots de passe mise en œuvre par les administrateurs de sécurité eux-mêmes doit, à notre sens répondre aux critères suivant :

- utiliser un matériel et un logiciel courant, c'est-à-dire des moyens accessibles à un agresseur peu fortuné ;
- être effectuée très régulièrement pendant une longue période (les variations sont nettement plus intéressantes qu'un résultat ponctuel) ;
- être effectuée sous des conditions de sécurité *très* strictes (ce qui n'est assez facile que si on isole les calculs sur une machine peu utilisée et si on prend garde à la transmission des données) sous peine de donner un très mauvais exemple ;
- les résultats doivent être consolidés - les indicateurs les plus pertinents n'étant pas toujours évidents à identifier en première approche (le ratio vulnérables/non-vulnérables est intéressant ponctuellement tandis que, dans une vision continue, l'évolution des taux d'apparition et de disparition de mots de passe vulnérables l'est probablement plus) ;
- la communication des résultats obtenus doit également être effectuée avec précaution, qu'il s'agisse de résultats ponctuels ou statistiques[8].

Parmi les logiciels utilisables pour réaliser ce type d'évaluation, les plus connus sont L0phtCrack (http://www.atstake.com/products/lc/) dans le domaine commercial pour la version 5 et plutôt sur plate-forme Windows ; et John the Ripper (http://www.openwall.com/john/) plutôt sur plate-forme Unix. Ces outils sont les successeurs de l'outil historique dans ce domaine, nommé : Crack (v4.1 en 1991 puis v5.0 (http://www.crypticide.com/users/alecm/security/c50-faq.html)) d'Alec Muffet (http://www.crypticide.com/dropsafe/info/home.html).

Dans le domaine des outils orientés vers la capture des mots de passe (chiffrés ou non) transitant sur le réseau, un outil librement disponible particulièrement intéressant est Caïn (http://www.oxid.it/cain.html). Il offre un niveau de convivialité remarquable pour une démonstration et un outillage intéressant pour divers types de mots de passe et d'écoute réseau.



Cain, un outil de capture passive et d'attaque des mots de passe

Annuaires

La problématique du stockage des mots de passe dans le système d'information est de plus en plus liée à celle de la mise en place d'un annuaire des utilisateurs. Quoique le mot de passe soit un attribut assez particulier de la définition des utilisateurs (en comparaison, par exemple, de leur nom ou de leur numéro de téléphone), sa gestion est étroitement associée à celle des autres caractéristiques de l'utilisateur.

Parmi les différentes technologies d'annuaires, certaines permettent de traiter de manière unifiée la gestion des mots de passe (NIS+, LDAP), d'autres s'appuient sur une infrastructure de gestion des informations et des fonctions d'authentification distincte (par exemple Active Directory et Kerberos), d'autres offrent des fonctions spécifiques (DNSSEC).

Le lien entre l'annuaire des utilisateurs offert par le système d'information et les informations concernant la sécurité pourrait être amené à s'étendre encore avec la gestion dans l'annuaire d'éléments comme les certificats X.509 des utilisateurs, ou les clefs et les certificats IPSEC/IKE pour les dialogues entre machines ou entre services.

SSO

Face à la multiplication des mots de passe gérés par un même utilisateur, des utilitaires de gestion sont apparus pour aider les utilisateurs à stocker leurs différents mots de passe et leur éviter d'avoir à les mémoriser. Ces différents utilitaires sont généralement rassemblés sous la désignation de moyens de « SSO » (pour *Single Sign-On*).

Cet acronyme désigne en fait de manière générique la problématique de l'authentification unique : c'est à dire le fait pour l'ensemble des applications et des systèmes d'exploitation d'un système d'information réparti de partager le(s) même(s) système(s) d'authentification et d'éviter ainsi aux utilisateurs d'avoir à s'authentifier plusieurs fois pour accéder aux différentes ressources du système d'information. Cette problématique, plus proche en fait de l'autorisation dans un système distribué que de l'authentification, a été abordée initialement dans le projet Kerberos (http://web.mit.edu/kerberos/www/) du MIT.

Toutefois, en l'absence de l'utilisation d'un même service d'authentification suffisamment sophistiqué pour permettre à l'ensemble du système de fonctionner avec une authentification unique, la plupart des applications utilisent en fait des annuaires et des définitions d'utilisateurs internes et exigent une ré-authentification des utilisateurs. Les utilitaires de SSO permettent de donner l'illusion d'une authentification unique en automatisant les authentifications « secondaires » à partir des informations stockées dans un répertoire de chaque utilisateur (qui contient les identifiants et les mots de passe nécessaires). Qui plus est, certains utilitaires permettent le stockage de ce répertoire dans une ressource centralisée du système d'information (comme Active Directory ou LDAP) ce qui complète l'illusion en associant l'accès à ce répertoire à l'authentification « primaire » assurée par les annuaires. D'autres utilitaires offrent la possibilité de stocker les données d'identification et d'authentification dans la mémoire protégée de dispositifs matériels comme les cartes à puce, ou les clefs USB ; ce qui permet à un utilisateur de transporter avec lui en permanence tout le répertoire dont il a besoin, et de mélanger les éléments personnels et professionnels (sur ce dispositif qui lui est propre).

L'utilisation de ces utilitaires simplifie grandement la vie aux utilisateurs, désormais confrontés quotidiennement à l'obligation d'utiliser 4 ou 5 mots de passe au moins. Ils sont donc associés à une demande qui peut être forte. Par ailleurs, ils semblent parfois indispensables pour permettre de prendre en charge des applications anciennes dont le développement est définitivement arrêté et qui ne pourront pas évoluer vers la prise en compte d'un nouveau système d'authentification. Une automatisation poussée et fiable permettrait même d'envisager une gestion complètement transparente des authentifications secondaires, et donc de choisir l'utilisation de mots de passe très complexes, impossibles à mémoriser pour un individu, et changeant fréquemment. Dans la pratique toutefois, un tel degré d'automatisation semble difficile à atteindre. Enfin, ils sont particulièrement adaptés à un usage personnel pour la gestion des multiples comptes des applications Web.

Toutefois, à notre sens, il faut considérer que ces utilitaires ne donnent qu'une illusion d'authentification unique. Pour y aboutir réellement, il faut permettre aux applications d'utiliser un service d'autorisation suffisamment évolué pour gérer un système réparti moderne (et si possible ouvert) ; le premier des services qu'il rend étant bien évidemment celui de permettre et de garantir l'authentification des utilisateurs (et des services) concernés auprès de lui. Toutefois, tant que les solutions disponibles sur ce point n'auront pas connu un déploiement effectif suffisamment étendu, les utilitaires de SSO de gestion des mots de passe semblent avoir un bel avenir devant eux.

La plupart des systèmes d'exploitation ou des éditeurs de logiciels de sécurité à destination du poste de travail (fournisseurs d'antivirus par exemple) proposent désormais ce type de logiciel. On peut ainsi mentionner : KDE Wallet Manager, GNOME Password Manager, Symantec's Norton Password Manager, Aladdin eToken Web Sign-On ou même certaines fonctions d'Internet Explorer et de Mozilla ou Firefox, et bien d'autres.

Notes

1. Cette zone ne doit pas pouvoir être accédée, même par une intrusion physique sur la puce, sans provoquer la destruction de la mémoire concernée.
2. Avant l'introduction des fichiers `/etc/shadow` contenant les mots de passe cryptés accessibles seulement à certaines catégories d'utilisateurs, ceux-ci étaient directement stockés dans la table `/etc/passwd` accessible en lecture par tous les utilisateurs.
3. Les cartes à puce récentes permettant désormais d'utiliser n'importe quelle séquence alphanumérique pour choisir un PIN et les certificats X.509 pouvant être entièrement gérés par une carte à puce, la distinction entre PIN et passphrase commence également à s'estomper.
4. Au contraire du mot de passe de paramétrage de l'équipement (*enable password*), généralement stocké avec la fonction de hachage sécurisée MD5, la plupart des autres mots de passe sont stockés sous une forme inversible pour une discussion factuelle du sujet. Voir http://www.auscert.org.au/render.html?it=285
5. http://lasecwww.epfl.ch/pub/lasec/doc/Oech04.pdf
6. http://lasecwww.epfl.ch/
7. Des réactions négatives sont à craindre si on montre sans précaution aux utilisateurs que leurs mots de passe sont « mauvais » (faillibles en fait) d'autant qu'aucune méthode de sélection d'un mot de passe n'est infaillible, surtout face aux progrès de la recherche exhaustive.
8. Il est *bien entendu* totalement inadapté de provoquer les utilisateurs dont le mot de passe a été identifié en le leur dévoilant sans sollicitation. De toute façon, la curiosité dans ce domaine est un vilain défaut.

Sécurité informatique/Chiffrement de flux et VPN

La protection des flux réseaux, et notamment des flux point à point identifiables facilement, comme les transferts de fichiers entre entreprises ou les accès des ordinateurs portables des utilisateurs nomades vers le réseau interne d'une entreprise à partir d'Internet, peut bénéficier d'une technique de protection générique consistant à authentifier et chiffrer l'ensemble du flux réseau concerné. Les technologies permettant cette protection sont souvent regroupées sous la désignation de « VPN » pour Virtual Private Network (réseau privé virtuel). On peut grossièrement dissocier les implémentations entre :

- la mise en œuvre d'un tunnel réseau protégé directement au niveau IP entre deux machines qui protège tous les échanges effectués mais peut théoriquement contraindre l'une des machines (située dans un environnement hostile) à interrompre ses communications hors du VPN ;
- et la création d'un tunnel au niveau des services applicatifs (généralement au niveau TCP) qui peut permettre de protéger plus spécifiquement les flux associés à un service donné (les flux X11 par exemple, voire les flux HTTP via SSL par exemple si on veut bien voir HTTPS comme un tunnel).

Suivant que l'on souhaite protéger les communications au niveau IP ou TCP, la problématique d'authentification est assez différente. Dans le premier cas, il s'agit de réaliser une authentification mutuelle des machines détentrices des adresses IP concernées. Dans le deuxième cas, l'authentification peut éventuellement également porter sur les utilisateurs des applications activant le tunnel. Cette frontière n'est pas hermétique, mais il faut apparemment utiliser des extensions des techniques de VPN de niveau IP pour incorporer une authentification utilisateur quand elle est souhaitée, par exemple pour des accès nomades (voir IPSEC).

Dans les deux cas, la mise en œuvre fait appel à des techniques assez similaires : authentification forte (par exemple à l'aide de certificats X.509, ou des algorithmes d'authentification basés sur RSA ou DSA), négociation de clés intermédiaires et d'algorithmes de chiffrement (DES, 3DES, AES, Blowfish, etc.) et/ou de contrôle d'intégrité (MD5, SHA-1, SHA-256, etc.) et traitement du flux avec mise à jour régulière des clés. On trouve donc un grand nombre de points communs entre les deux approches. Par ailleurs, dans le domaine des standards normalisés et des implémentations les plus répandues, deux grands acteurs dominent la mise en œuvre : IPSEC/ISAKMP pour le chiffrement au niveau IP, et SSL (via SSH ou HTTPS) pour le chiffrement au niveau des flux applicatifs TCP.

IPSEC

Les objectifs de l'architecture IPsec présentée dans la [RFC 2401] (sur laquelle cette section est largement basée) sont de fournir différents services de sécurité pour le trafic au niveau IP, que ce soit pour les environnements IPv4 ou IPv6, via l'utilisation de mécanismes de sécurité cryptographiques ou protocolaires. Les différents éléments composant cette architecture de sécurité sont :

- les protocoles de sécurité : AH (*Authentication Header*) pour la garantie de l'intégrité, et ESP (*Encapsulating Security Payload') pour la confidentialité ;*
- les associations de sécurité (SA) : leur définition, leur fonctionnement, leur gestion, et les traitements associés ;
- la gestion des clefs de chiffrement : manuelle ou automatique, et notamment via IKE (*Internet Key Exchange*) ou dans le schéma général d'ISAKMP ;
- et enfin les algorithmes de protection de l'intégrité ou de chiffrement eux-mêmes ([RFC 2403], [RFC 2404], etc.).

IPsec fournit les services de sécurité en permettant à un système informatique de sélectionner les protocoles de sécurité souhaités, de déterminer le(s) algorithme(s) à utiliser pour ces services, et en mettant en place les clefs cryptographiques nécessaires pour leur mise en œuvre.

L'ensemble des services de sécurité fournis par IPsec incluent le contrôle d'accès, la protection de l'intégrité des paquets, la garantie d'authenticité de l'origine des paquets, le rejet des paquets rejoués (c'est à dire une forme de protection partielle d'intégrité sur les séquences de paquets), la confidentialité (via le chiffrement), et une confidentialité partielle sur la nature des flux réseaux transportés. Comme ces services sont fournis au niveau IP, ils peuvent être utilisés par n'importe quel protocole de plus haut niveau, comme TCP, UDP, ICMP, BGP, etc.

IPsec supporte aussi la négociation de la compression au niveau IP, notamment en raison du fait que, quand le chiffrement est employé, il empêche toute compression efficace par les protocoles de plus bas niveau.

IPsec

IPsec utilise deux protocoles pour assurer la sécurité du trafic réseau : AH et ESP. Ces protocoles sont décrits respectivement dans les [RFC 2402] et [RFC 2406].

- Le mode AH (*Authentication Header*) fournit une protection de l'intégrité des paquets, la garantie de l'origine des paquets et une protection optionnelle contre les rejeux.
- Le mode ESP du protocole (*Encapsulating Security Payload*) fournit la confidentialité. Il peut aussi fournir une protection de l'intégrité des paquets, la garantie de l'origine des paquets et une protection contre les rejeux.
- AH et ESP sont tous deux des véhicules pour le contrôle d'accès, basé sur la distribution de clefs cryptographiques et la gestion des flux réseaux associés aux protocoles de sécurité.

Ces protocoles peuvent être utilisés seuls ou simultanément pour fournir des services de sécurité sur IPv4 ou IPv6. Chaque protocole permet d'utiliser deux modes : le mode transport ou le mode tunnel. Dans le mode transport, ils fournissent essentiellement une protection à l'usage des protocoles de plus haut niveau ; dans le mode tunnel, ces protocoles sont appliqués à des paquets IP encapsulés.

IPsec permet donc à l'utilisateur ou l'administrateur de contrôler la granularité du trafic auquel un service de sécurité est offert. Par exemple, il est possible de créer un seul tunnel crypté pour transporter tout le trafic réseau entre deux passerelles sécurisées ou bien un tunnel chiffré séparé peut être construit pour chaque connexion TCP établie entre chaque paire de machines communiquant au travers de ces passerelles. Pour permettre cette flexibilité, l'infrastructure de gestion d'IPsec doit inclure des fonctions permettant :

- de spécifier quels services de sécurité doivent être utilisés et comment ils doivent être combinés ;
- de définir la granularité à laquelle un niveau de protection donné doit être appliqué ;
- de choisir les algorithmes utilisés concrètement pour les protections cryptographiques.

Comme tout ces services de sécurité reposent sur des valeurs secrètes partagées (des clefs cryptographiques), IPsec repose sur un ensemble de mécanismes séparés nécessaires pour mettre en place ces clefs. IPsec supporte à la fois une distribution manuelle ou automatique des clefs de chiffrement, d'intégrité et d'authentification. L'ensemble des normes relatives à IPsec fournit une solution spécifique pour la gestion automatisée des clefs : IKE, basée sur des solutions de cryptographie asymétrique (à clef publique), mais d'autres approches peuvent être employées, en s'appuyant sur le cadre général fourni par ISAKMP.

ISAKMP/IKE

ISAKMP, défini dans la [RFC 2408], est le protocole permettant la mise en place des associations de sécurité (SA) utilisables pour la mise en œuvre du tunnel chiffré. Ce protocole ne définit pas précisément les techniques d'authentification et d'échange de clefs utilisables mais fournit le contexte permettant de définir ces techniques.

Le protocole technique le plus utilisé du point de vue opérationnel semble être IKE, défini dans la [RFC 2409], à l'intérieur de l'ensemble normatif d'IPsec. D'autres protocoles sont possibles mais moins répandus, par exemple OAKLEY (défini dans la [RFC 2412] et utilisant une technique de type Diffie-Helmman) dont IKE est une variante simplifiée.

Déroulement d'une session

Les différentes phases de l'établissement d'une session IPsec sont schématiquement les suivantes :

1. Un tunnel IPsec est initié lorsqu'un trafic à protéger devant aller d'un point à un autre est détecté (soit sur la machine elle-même en mode transport, soit par la passerelle du site en mode tunnel).
2. Phase 1 (IKE) : négociation de la politique d'établissement des associations de sécurité (SA) ISAKMP. Une fois que les 2 extrémités du tunnel sont authentifiées, un canal de communication protégé est créé pour la poursuite de la négociation IKE.
3. Phase 2 (IKE) : les 2 extrémités du tunnel utilisent alors ce canal protégé pour négocier les associations IPsec (ESP et/ou AH). La négociation des paramètres finaux détermine comment le tunnel IPsec établi pourra fonctionner (algorithmes utilisés, intervalles de renouvellement des clefs de chiffrement, etc.).
4. Le tunnel IPsec est créé, les échanges des données transférées entre les 2 extrémités du tunnel IPsec sont basées sur les paramètres IPsec configurés dans le « transform set » choisi.
5. Le tunnel IPsec se termine quand les SA sont supprimées ou quand leur durée de vie expire.

Les phases de négociation 1 et 2 sont cruciales pour l'établissement de la session IPsec. Surtout quand on a affaire à des implémentations hétérogènes de constructeurs différents, les critères de succès de la négociation peuvent être difficiles à atteindre. En effet, le protocole est assez complexe, quasiment impossible à observer sur le réseau (car chiffré dès les premières phases) et met en jeu un nombre important de paramètres parfois assez cryptiques. En pratique, la partie la plus difficile de la mise en œuvre d'un VPN IPsec consiste généralement à bien paramétrer les logiciels pour assurer le succès de cette négociation. Quand on a affaire à la même implémentation à chaque bout du tunnel, les fichiers de configuration sont presque les mêmes, et la tâche est généralement plus simple.

Pendant la durée de vie de la session de communication IPsec, des négociations périodiques sont effectuées via ISAKMP pour faire varier les clefs de sessions utilisées. En règle générale, les clefs de sessions utilisées pour les protocoles ESP ou AH (type phase 2) sont renégociées avec une période de l'ordre d'une heure, celles utilisées pour les négociations ISAKMP elles-mêmes au bout d'une période de l'ordre d'un jour. La valeur optimale de ces paramètres varie bien évidemment suivant les algorithmes cryptographiques autorisés (et notamment la longueur des clefs impliquées) et le volume de trafic échangé.

Clients VPN « personnels » (authentification de l'utilisateur)

Une utilisation particulièrement importante des protections de type VPN concerne les accès des utilisateurs nomades aux ressources du réseau interne d'une entreprise. Dans ce mode de fonctionnement, la protection accrue offerte par l'utilisation d'un VPN (à la fois en terme d'authentification et de protection du flux pendant son activité) permet d'envisager de permettre à ces postes de travail itinérants d'accéder à la majorité des ressources du réseau interne à partir d'Internet (et donc de points d'accès opérateurs classiques, avec une couverture géographique très large) voire à partir d'un point d'accès sans fil (*WiFi*).

Dans ce cas toutefois, on souhaite généralement associer l'authentification mise en œuvre par le logiciel VPN (par exemple une implémentation IPsec) avec une authentification de l'utilisateur accédant au réseau interne. Dans ce cas, l'authentification effectuée par un protocole comme IPsec n'est pas totalement adaptée. Par ailleurs, le souci pratique est fréquemment de ré-utiliser une infrastructure d'authentification existante pour les utilisateurs (utilisant par exemple S/Key, ou de simples mots de passe via RADIUS) et de ne profiter que de la sécurité offerte par IPsec au niveau du transport (pas tellement au niveau de l'authentification).

Il faut aussi souligner, dans ce cas, l'importance de la protection de l'ordinateur portable lui-même par rapport aux ressources réseau qui peuvent éventuellement l'entourer. En effet, la liaison VPN offrant généralement un accès au niveau IP, même si les flux sont ultérieurement contrôlés par un firewall de l'entreprise, il peut être possible de rebondir vers le réseau interne en prenant appui sur l'ordinateur portable si celui-ci est vulnérable. C'est tout à fait possible par exemple en activant les fonctions de routage du système d'exploitation de l'ordinateur portable, ou en utilisant des relais applicatifs ou des relais génériques (logiciels que nous avons déjà présentés sous un autre éclairage). Ceci conduit généralement à coupler l'utilisation d'un « client » VPN avec un firewall personnel permettant d'interdire tout accès réseau n'empruntant pas le canal protégé par le VPN. Il faut noter que, notamment pour pallier aux attaques les plus complexes ou aux infections virales capables de se propager en différé, les protections du firewall personnel doivent aussi être actives même quand le VPN lui-même n'est pas actif.

Face à ces besoins, même si le fonctionnement d'IPsec est souvent respecté pour tout ce qui est du fonctionnement courant du VPN, un certain nombre d'implémentations « propriétaires » ont vu le jour. (En toute rigueur, un certain nombre d'implémentations ont aussi précédé l'apparition des standards IPsec.) De manière générale, ces clients VPN « personnels » propriétaires offrent des fonctionnalités additionnelles en terme d'authentification ou de protection mais ces extensions ne respectent généralement pas une norme et sont donc généralement uniquement compatibles avec une passerelle du même constructeur. Dans la pratique, il est important de les identifier correctement, que ce soit pour profiter de ces fonctionnalités additionnelles ou pour ne pas en pâtir.

Tunnels SSH

Pour la mise en place de liaisons protégées de type VPN au niveau applicatif, une solution pragmatique qui permet parfois de trouver des solutions efficaces dans les cas où le mode de fonctionnement d'IPsec est un peu trop contraignant, est d'utiliser le mode tunnel de connexions SSH.

SSH est d'abord un outil de connexion à distance de machine à machine offrant une authentification forte (par chiffrement asymétrique RSA ou DSA en général, éventuellement couplé à une authentification par mot de passe Unix classique) et une protection de la session distante (à la fois du point de vue de sa confidentialité et de son intégrité). Toutefois, le protocole SSH permet également d'utiliser le canal de communication TCP protégé pour transporter les communications réseau d'autres applications (si celles-ci le permettent sans trop de complication). Cette configuration est notamment recommandée pour renforcer la sécurité de sessions X11 distantes traversant des réseaux non-sûrs, en utilisant les fonctions de *x-forwarding* offertes à la fois par OpenSSH (http://www.openssh.org/) et XFree86 (http://www.xfree86.org/)/XOrg (http://xorg.freedesktop.org/).

L'avantage de cette solution est de pouvoir être mise en place éventuellement entièrement en mode utilisateur, en tout cas sans impliquer des paramétrages sophistiqués impliquant potentiellement l'ensemble du fonctionnement réseau des machines concernées.

OpenVPN

Une solution similaire mais plus récente est dédiée à la mise en place de tunnels au niveau applicatif. Il s'agit d'OpenVPN (http://openvpn.sourceforge.net/). Elle utilise également les possibilités offertes par le protocole TLS et son implémentation OpenSSL. Cette réalisation illustre bien les avantages de fonctionner au niveau applicatif par rapport à IPsec : OpenVPN fonctionne sur la majeure partie des systèmes d'exploitation et peut traverser assez facilement des firewall intermédiaires. Cette utilitaire qui entre en version 2 nous semble particulièrement intéressant.

Sécurité informatique/Détection d'intrusion

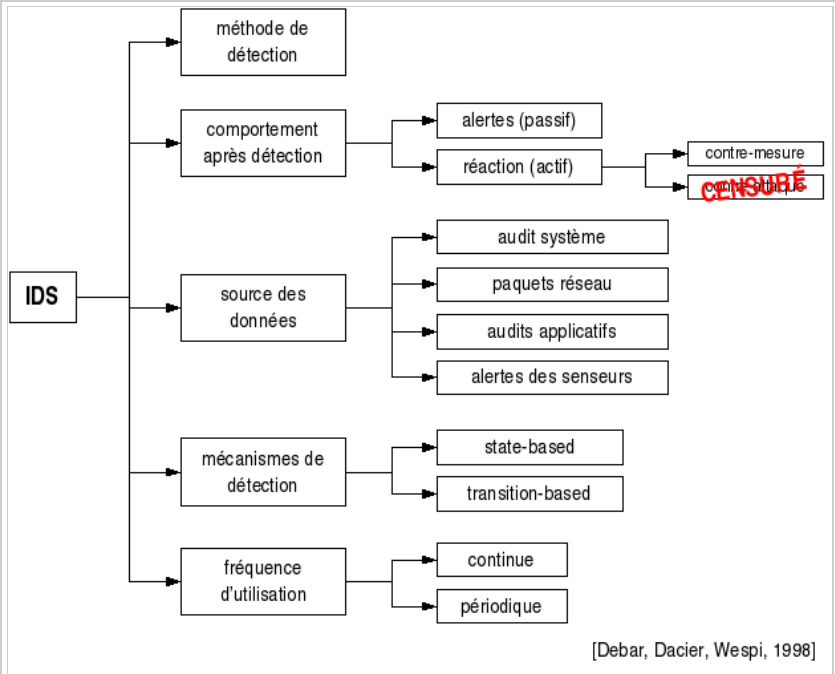
Par rapport aux moyens présentés précédemment qui sont avant tout des moyens de protection ou prévention des intrusions, les techniques de détection d'intrusion sont orientées vers la surveillance du système informatique et constituent dans une certaine mesure des moyens de tolérance aux intrusions.

L'objectif de la détection d'intrusion est de repérer les actions d'un attaquant tentant de ou tirant partie des vulnérabilités du système informatique pour nuire aux objectifs de sécurité du système, ou utilisant des techniques recensées comme des techniques d'attaque. La relation entre une intrusion et les objectifs de sécurité du système n'est pas toujours mentionnée, mais nous considérons qu'il est indispensable de se référer à la politique de sécurité du système informatique pour définir précisément ce qui est une intrusion pour un système donné, c'est à dire une défaillance de sécurité.

Il est vrai que, dans l'état de l'art au sens le plus large, il est possible de recenser un certain nombre d'actes ou d'actions techniques qui sont considérées par une large majorité comme des attaques répréhensibles. Dans ce cas, un système détectant l'occurrence de ces attaques peut suffire à identifier des intrusions sans établir de relation directe avec les objectifs de sécurité. Toutefois, dans nombre de situations concrètes, c'est plutôt la situation inverse qui est rencontrée : ce sont des actions suspectes mais dont la malveillance n'est pas avérée qui sont identifiées. C'est alors bien souvent la mise en relation avec les objectifs de sécurité du système qui permet de décider de l'intérêt de suivre ou non ces actions détectées et leurs conséquences. Par ailleurs il nous semble que la politique de sécurité de tout système d'information se doit aussi d'interdire ou tout au moins d'encadrer dans un cadre obligatoire précis l'exécution d'attaques (au sens commun) sur le système informatique.

Terminologie

La caractérisation des différents systèmes de détection d'intrusion existants explorés dans le domaine de la recherche a conduit à la classification terminologique présentée dans la figure suivante. Cette terminologie permet de différencier les systèmes de détection d'intrusion (ou IDS pour *Intrusion Detection System*) suivant un certain nombre de caractéristiques.



Terminologie de la détection d'intrusion

On peut d'abord faire une distinction assez fondamentale sur la méthode de détection utilisée par l'IDS. Il existe deux grandes catégories de méthodes de détection explorées dans la littérature : celles basées sur une approche comportementale (par exemple l'analyse statistique, l'analyse bayésienne, les réseaux neuronaux) et celles basées sur une approche par scénarios (par exemple la recherche de signatures, le pattern matching, ou la simulation de réseaux de Petri par exemple).

Globalement, les approches comportementales visent à reconnaître un comportement anormal, que ce soit par rapport à une définition du comportement normal ou anormal fournie au système de détection d'intrusion (par exemple une spécification de protocole de communication) ou par rapport à une modélisation des comportements normaux ou anormaux apprise à partir d'une observation préalable du système (en salle blanche, ou tout simplement en réel). Dans le cadre d'une approche comportementale, l'apprentissage semble donc possible, tout comme la possibilité de détecter des attaques inconnues au moment de la conception de l'IDS, à condition qu'elles génèrent des anomalies perceptibles dans le fonctionnement normal.

Par contre, dans une approche par scénarios, l'IDS s'appuie sur une base de connaissance pré-existante décrivant les comportements normaux ou anormaux et utilise cette connaissance pour la reconnaissance des évènements produits par des actions d'intrusions dans le système informatique qu'il observe. Cette méthode implique donc la constitution et la mise à jour régulière d'une base de connaissance référençant les différentes attaques connues susceptibles d'être mises en œuvre dans un système informatique. C'est à partir de ces informations, affinées par l'administrateur en fonction du système surveillé, que l'IDS identifie d'éventuelles attaques ayant lieu dans le système informatique. Dans cette approche, l'IDS se focalise donc sur l'identification des utilisations abusives (misuse). Une autre mise en œuvre conforme à la terminologie mais originale dans la pratique de cette approche de détection par scénarios consiste à constituer une base de connaissance des comportements permis dans le système (et non des comportements abusifs) pour configurer les actions de détection (des utilisations normales en quelque sorte).

On peut ensuite comparer les systèmes de détection d'intrusion en fonction du mode de fonctionnement des mécanismes de détection qu'ils mettent en œuvre. De manière générale, un IDS peut tenter d'identifier des attaques en s'appuyant sur des informations relatives aux transitions ayant lieu dans le système (l'exécution de certains programmes, de certaines séquences d'instructions, l'arrivée de certains paquets réseau, etc.) ou bien en étudiant l'état de certaines parties du système (par exemple, l'intégrité des programmes stockés, les privilèges des utilisateurs, les transferts de droits, etc.).

Ensuite, les IDS s'appuient généralement sur des sources de données différentes. Certains IDS, dits « réseau » ou NIDS (pour *Network IDS*) examinent les paquets transportés par le réseau. D'autres IDS, dits « système » ou HIDS (pour *Host IDS*) s'appuient sur les traces d'exécution fournies en général par le système d'exploitation mais parfois aussi par les applications. (Bien que la distinction ne soit pas très marquée dans la pratique, il est utile de distinguer l'audit de bas niveau disponible au niveau des systèmes d'exploitation (qui peut aller jusqu'à des traces listant les appels systèmes exécutés) des informations d'audit de plus haut niveau,

généralement plus riches de sens mais moins exhaustives, fournies par certaines applications.) D'autres IDS sont également en train d'apparaître appuyés sur des fonctions de corrélation et, dans ce cas, on peut considérer qu'il sont eux-mêmes des systèmes utilisant les alertes d'autres senseurs comme source de données.

Lors de la détection d'une attaque, un système de détection d'intrusion peut adopter plusieurs comportements. En règle générale, une réponse passive est adoptée : l'IDS diffuse une alerte identifiant l'attaque détectée vers un système d'analyse ou de diffusion. Toutefois, on peut envisager des réponses plus actives, parmi lesquelles nous distinguons d'abord la mise en place (automatique) de contre-mesures destinées à limiter la portée d'une intrusion éventuelle. Par exemple, l'IDS peut réagir en reparamétrant un firewall pour mettre en place des règles de blocage temporaires de certains flux réseaux anormaux. Bien entendu, il importe de bien valider la fiabilité de la détection d'intrusion avant d'activer de telles contre-mesures automatiques. Dans la pratique, peu d'administrateurs sécurité envisagent concrètement de mettre en place ce type de réaction. Nous laissons au lecteur le soin de deviner la dénomination d'une réponse active après détection d'une attaque - duale d'une contre-mesure dans le vocabulaire « tactique ». Dans la pratique, nous espérons que peu d'administrateurs chargés de la sécurité envisagent cette dernière catégorie de comportement : compte tenu de la législation française, ce type de réaction ne nous semble pas permis dans le domaine civil.

Enfin, on peut également intégrer la fréquence d'utilisation de l'IDS dans les caractéristiques identifiables. Dans la perception courante d'un IDS, celui-ci est toujours utilisé de manière continue. Toutefois, dans certains cas, il peut également être important de bien identifier une détection effectuée en temps différé. (Par exemple du fait d'un HIDS analysant des fichiers traces transmis seulement toutes les heures.) On peut aussi envisager que certains outils dont l'utilité est avérée dans la pratique y compris pour révéler les traces d'une intrusion passée trouvent leur place dans cette classification au niveau d'une fréquence d'utilisation périodique, comme les outils de vérification d'intégrité (type Tripwire) ou les outils de recherche de vulnérabilité (type COPS, SATAN, ou Nessus).

Alertes et fausses alertes

Parmi les comportement possibles pour un IDS, on peut envisager les quatre possibilités recensées dans le tableau suivant qu'une intrusion est ou non en cours dans le système informatique et que le système de détection d'intrusion a émis ou non une alerte.

Comportements envisageables pour un IDS

	Pas d'alerte	**Alerte**
Pas d'attaque	Vrai négatif	Faux positif
Attaque en cours	Faux négatif	Vrai positif

Parmi ces quatre comportements, les vrai négatif et vrai positif correspondent aux comportements souhaités. Toutefois un IDS est généralement imparfait et conduit à l'apparition des deux autres comportements non désirés. Parmi eux, un faux négatif correspond à une attaque non détectée, et un faux positif à l'émission d'une fausse alerte. Les différents IDS souffrent généralement d'imperfections donnant lieu à l'apparition de ces comportements non désirés, mais selon des axes différents suivant les méthodes de détection qu'ils utilisent.

Un reproche fréquemment fait en direction des IDS utilisant une méthode de détection comportementale est de contenir dans leur principe même de fonctionnement la possibilité de fausses alertes (un changement de comportement légitime détecté comme anormal) ou de faux négatifs (par exemple pour une attaque très lente) ; tandis que les approches par scénarios semblent théoriquement être plus exactes. Toutefois, la base de connaissance utilisée dans les IDS par scénarios exige une maintenance constante et, dans la pratique, souffre également nécessairement d'imperfections. Malgré tout, la réputation des IDS comportementaux semble avoir souffert durablement de leur imperfection de principe (à notre sens de manière assez injustifiée), notamment du fait de la possibilité de faux négatifs.

Bien que les faux négatifs soient effectivement le premier des comportements indésirables pour un IDS, les faux positifs sont importants aussi : ils peuvent conduire à une réelle perte de confiance dans les capacités de détection de l'IDS de la part des administrateurs qui peut finir par remettre en cause la finalité de l'IDS. C'est même une des voies d'attaque envisageables contre un système équipé d'un IDS : générer un nombre suffisamment important de fausses alertes pour réduire l'attention des administrateurs et dissimuler une attaque réelle. De plus, dans la pratique, les faux positifs dus à l'environnement de l'IDS ou à des signatures d'attaque un peu trop affirmatives sont souvent nombreux ; et ceci nécessite généralement un reparamétrage de l'IDS pour faciliter son exploitation, au prix de l'introduction de possibilités de faux négatifs. La gestion des faux positifs est le premier problème auxquels sont confrontés les administrateurs d'un IDS, et il est généralement de taille. A notre sens, les IDS basés sur une approche par scénarios, c'est à dire la plupart des IDS courants, souffrent sur ce point d'un réel problème qui demanderait certainement de développer à la fois les possibilités d'adaptation de l'IDS à son environnement (peut-être par des moyens de corrélation) et une meilleure validation des signatures d'attaque disponibles.

L'utilisation de techniques de corrélation d'alertes provenant de plusieurs IDS semble être une des voies envisageables pour traiter ces problèmes d'analyse des alertes et notamment des fausses alertes. Dans ce cadre, la diversification des méthodes de détection utilisées par les différents IDS, ainsi que de leurs sources de données est aussi à nouveau envisageable. (Dans un certain sens, il s'agit d'ailleurs de ré-inventer la roue une fois de plus puisque le précurseur des systèmes de détection d'intrusion, nommé IDES, combinait déjà l'utilisation d'une approche comportementale -statistique- et d'une approche à base de règles -système expert-, dans les années 1980 du côté de Stanford.)

Approches étudiées et tendances

Les différentes approches de la détection d'intrusion présentées dans la terminologie ont pour la plupart toutes été étudiées au niveau de système expérimentaux, notamment dans les laboratoires de recherche ou les universités.

Toutefois, à cette diversité des expérimentations s'oppose une focalisation quasi exclusive des solutions techniques disponibles dans le domaine industriel sur les systèmes de détection d'intrusion réseau utilisant une approche par scénarios, et plus précisément de signatures d'attaques. La quasi-totalité des IDS commerciaux utilisent cette approche, peut-être parce que c'était l'une de celles qui étaient les plus directes à mettre en œuvre et à déployer dans les systèmes informatiques courants. Des solutions existent également pour la détection d'intrusion à partir des traces systèmes, toujours en utilisant généralement une approche par scénario (recherche d'évènements dans les traces). Toutefois, la plupart de ces HIDS utilisent les traces standards des systèmes d'exploitation, généralement assez peu exhaustives (en tout cas en comparaison des fonctions d'audit système disponibles sur les systèmes C2 et supérieurs dans la classification du livre orange - ceci dit celles-ci absorbent une part significative des ressources d'une machine seulement dans l'objectif de surveiller l'autre part et sont rarement déployées).

Concrètement, ce sont donc les systèmes de détection d'intrusion réseau utilisant des bases de signatures qui dominent les mises en œuvre opérationnelles.

Schéma d'architecture IDWG

Plusieurs schémas ont été proposés pour décrire les composants d'un système de détection d'intrusion. Parmi eux, nous avons retenu celui issu des travaux de l' *Intrusion Detection exchange format Working Group* (IDWG (http://www.ietf.org/html.charters/idwg-charter.html)) de l' *Internet Engineering Task Force* (IETF (http://www.ietf.org/)) comme base de départ, car il résulte d'un large consensus parmi les intervenants du domaine. L'objectif des travaux du groupe IDWG est la définition d'un standard de communication entre certains composants d'un système de détection d'intrusion. Ce standard de communication s'appuie sur le modèle d'architecture présenté dans la figure suivante.

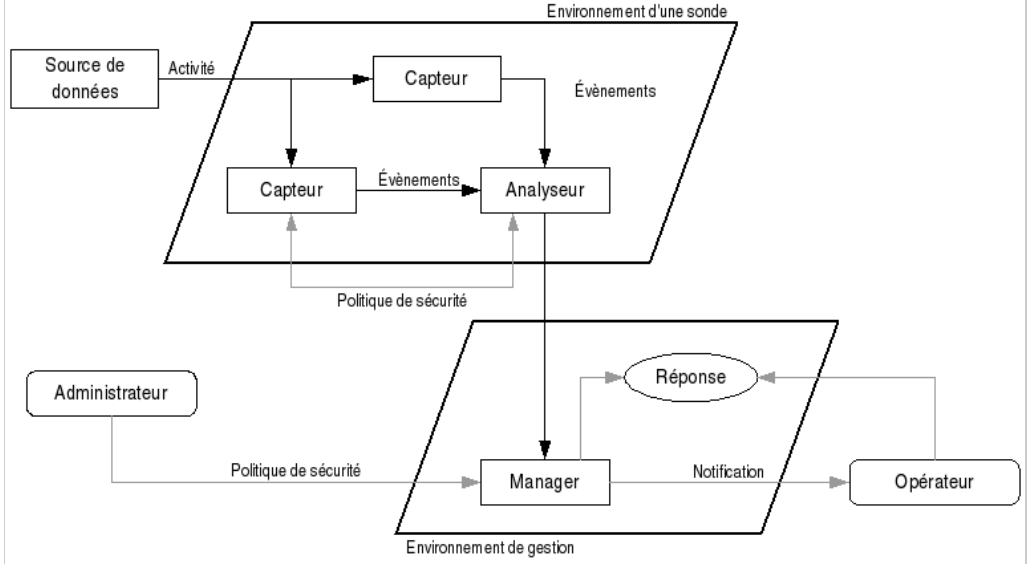
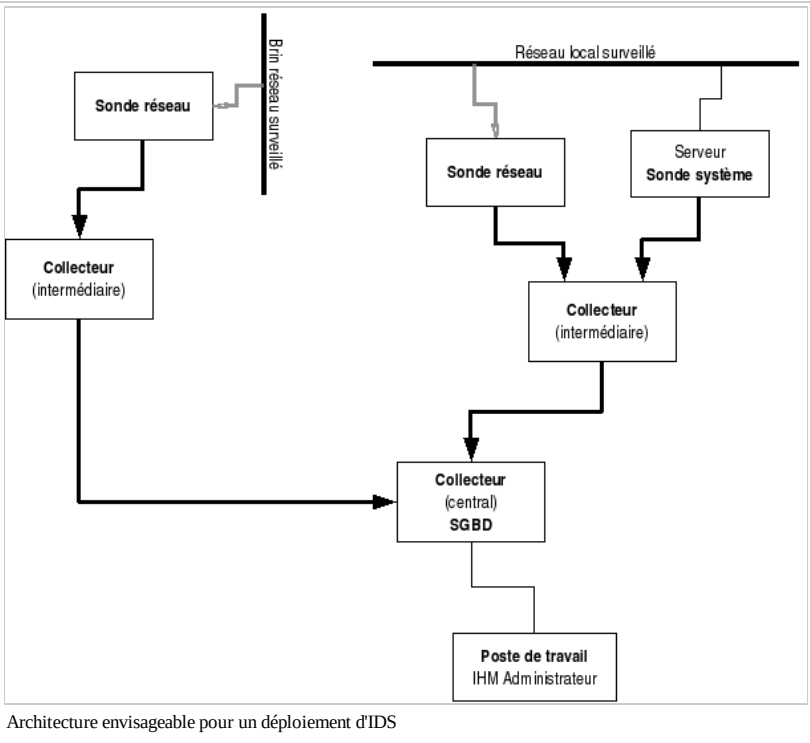


Schéma d'architecture IDWG d'un IDS

L'architecture IDWG contient des capteurs qui envoient des évènements à un analyseur. Un ou des capteurs couplés avec un analyseur forment une sonde. Une sonde envoie des alertes vers un manager qui la notifie à un opérateur humain.

Exemple d'architecture

Dans la pratique, le modèle fonctionnel d'architecture présenté précédemment doit être adapté pour une mise en place concrète des différents éléments techniques du système de détection d'intrusion. Nous présentons dans la figure suivante un exemple de déploiement de système de détection d'intrusion qui nous semble un peu plus représentatif des contraintes réelles de mise en place.



Architecture envisageable pour un déploiement d'IDS

Dans ce schéma, nous faisons apparaître trois sondes de détection d'intrusion : deux sondes réseau et une sonde système. On peut par exemple supposer que les deux sondes réseaux sont déployées à des endroits différents du système informatique l'une au niveau du réseau interne, l'autre par rapport à un point réseau intéressant (on peut aussi les envisager séparées sur des sites distants les uns des autres ou par des firewall). La sonde système peut par exemple être associée à un serveur de centralisation de traces mis en place sur le réseau local. Dans la mise en place présentée sur la figure, des collecteurs intermédiaires ont été ajoutés à proximité de deux groupes de sondes : la sonde réseau surveillant le brin réseau distant, et les deux sondes réseau et système s'intéressant au réseau interne. Ces deux collecteurs intermédiaires propagent ensuite les traces vers un collecteur central associé à un SGBD sur lequel les administrateurs de sécurité peuvent consulter et traiter les alertes.

Dans la pratique, ces collecteurs intermédiaires n'apparaissent généralement pas explicitement. Ils correspondent plus ou moins aux capacités de stockage temporaire et de transmission en différée des sondes vers leur manager. Le collecteur central correspond lui au manager des différentes sondes et offre à la fois des fonctions de gestion des sondes et de consultation des alertes. Toutefois, à notre sens il est intéressant de les faire apparaître explicitement pour mettre en évidence certaines des fonctions qui peuvent être utiles au niveau de la collecte et du transport des alertes. Les collecteurs intermédiaires peuvent gérer les flux réseaux et optimiser le transport des alertes en tenant compte, par exemple, des problèmes de disponibilités du collecteur central.

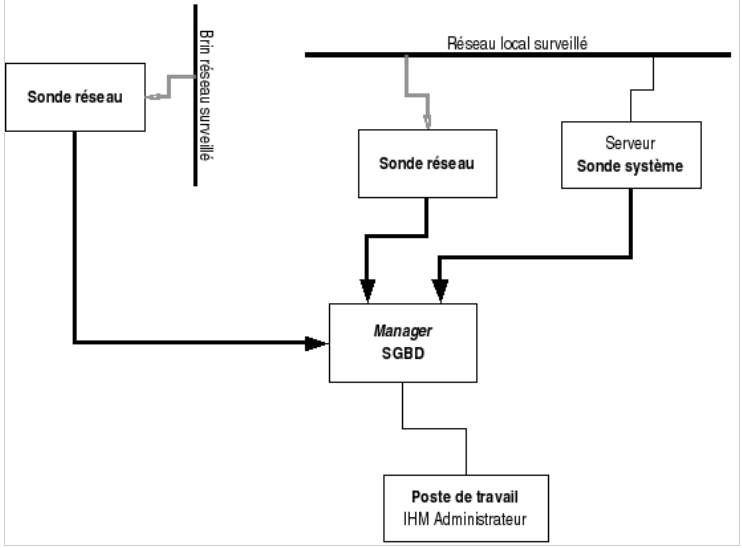
Si des fonctions de corrélation ou de groupement d'alertes sont ajoutées dans l'architecture de sécurité, celles-ci peuvent éventuellement être mises en œuvre de manière distribuée au niveau des collecteurs intermédiaires. Dans une architecture comme celle définie par l'IDWG (voir figure) ou dans la plupart des architectures

disponibles à l'heure actuelle, l'ensemble des traitements de corrélation doivent être mis en œuvre au niveau du collecteur central (manager associé éventuellement à un SGBD). Ce point central peut alors devenir un point critique du point de vue des performances. Par ailleurs, certains des traitements de corrélation simples (comme le groupement d'alertes) peuvent avantageusement être réalisées au plus près des sondes, en tirant ainsi parti d'une distribution des calcul. L'alternative à l'introduction (au moins au niveau logique) des collecteurs intermédiaires est alors de modifier le fonctionnement des sondes, ce qui pose aussi des problèmes de performance.

Les collecteurs intermédiaires peuvent aussi être associés plus étroitement à la problématique de gestion opérationnelle des sondes (lesquelles peuvent provenir de constructeurs différents) voire à la mise en œuvre de la réaction.

D'un point de vue décisionnel, il nous semble que ces réflexions ne se limitent pas à l'aspect de la collecte des alertes (ou même de la corrélation), l'introduction de ce niveau intermédiaire entre les sondes d'analyse et le manager correspond en quelque sorte à l'introduction d'un niveau décisionnel tactique entre les agents opérationnels (en contact direct avec le terrain) et un niveau d'analyse stratégique auquel les décisions générales d'action peuvent être prises avec plus de recul.

Par contre, dans la pratique, les architectures mise en oeuvre correspondent classiquement à la configuration présentée dans la figure suivante (très proche de celle identifiée par l'IDWG).



Architecture courante pour un IDS

Solutions (réseau)

Comme nous l'avons déjà mentionné, la plupart des solutions existantes en matière de détection d'intrusion concernent la mise en place de NIDS (IDS réseau) complétés éventuellement par certains HIDS (IDS système) et du logiciel de gestion associé (manager). Dans cette section, nous étudierons certaines des solutions les plus populaires dans les domaines commerciaux et open-source, à savoir l'IDS RealSecure et le NIDS Snort. Nous nous intéresserons également à une solution open-source complète, qui capitalise d'ailleurs sur certains des succés de Snort, le système Prelude-IDS.

RealSecure

RealSecure est la solution IDS proposée par la société ISS (Internet Security Systems). Cette solution s'appuie assez naturellement sur une architecture du type de celle proposée par l'IDWG. Dans sa solution, ISS propose toutefois deux types de sondes : un NIDS RealSecure Network Sensor (le plus connu), et un HIDS RealSecure Server Sensor. Ces sondes sont complétées par une machine de management (qui peut être installée au côté d'une sonde dans une configuration mono-machine) et un logiciel d'administration graphique de bonne qualité. La console de RealSecure est divisée en 3 grandes partie : la partie inférieure indique l'état des différentes sondes installées dans le système (et notamment leur niveau de mise à jour), la partie gauche rassemble les différentes alertes remontées par les sondes en temps réel classées selon le type de l'attaque détectée, la partie droite présente la même information mais classée en fonction de trois niveaux de gravité pré-établis pour les alertes.

Le NIDS utilise une base de signatures de reconnaissance élaborée par le centre de veille d'ISS, nommé la X-Force. ISS a développé un mécanisme spécifique de diffusion des mises à jour de cette base de signatures, communes à un certain nombre de ses produits, sous le nom d'eXpress Update (ou XPU). Ceci permet un déploiement rapide des mises à jour, auquel il faut pourtant ajouter une phase d'administration (pour l'instant indispensable) pour l'activation ou non des nouvelles signatures dans la configuration (ou « politique ») de sécurité des produits.

RealSecure propose également des moyens de réaction permettant d'envisager la mise en place automatique de contre-mesures en cas de détection d'une attaque, en reparamétrant certains modèles de firewall (CheckPoint notamment) pour bloquer les adresses IP à l'origine des attaques identifiées. L'outil permet d'effectuer un certain nombre de recherches dans la base des alertes collectées, ainsi que la production de rapports de synthèse concernant l'activité générale du système de détection d'intrusion.

Dans l'ensemble, la technologie fournie par RealSecure est assez complète, ce qui permet à ISS de proposer des déploiements de grande envergure, du type de ceux présentés dans ses brochures commerciales. Toutefois dans la pratique, RealSecure présente comme beaucoup d'IDS le problème de la gestion des nombreuses fausses alarmes générées par les sondes (et notamment les sondes réseau). Dans le cas de ce produit, c'est parfois assez frappant, au point de remettre en cause certains signatures, ou en tout cas en obligeant à une désactivation quasi-systématique. De la même manière, les fonctions d'analyse disponibles sont essentiellement tournées vers une finalité de production de rapports (comptage, regroupement par types pré-définis, par source ou par destination, etc.) et permettent assez difficilement d'appliquer une méthode même manuelle de corrélation entre les alertes pour aboutir à un diagnostic plus élaboré. Cette tâche est encore compliquée par le bruit de fond existant sur le réseau (il y a toujours, quelque part, un équipement ou un logiciel qui pollue la ligne avec un trafic certes suspect, mais généralement sans danger), révélé par les sondes de détection d'intrusion, mais auxquels, parfois, il n'est pas possible d'appliquer facilement une correction technique et dont on souhaiterait pouvoir automatiser le traitement des alertes associées. A l'usage, le système reste donc assez imparfait à moins de disposer de moyens importants permettant d'agir à la fois sur les sondes et sur les systèmes observés, ou de disposer d'un outil tiers d'analyse et de corrélation. ISS propose à présent dans sa gamme un produit nommé Site Protector, visant à répondre à ces problématiques d'analyse ; mais les fonctions de corrélation attendues semblent pour l'instant relativement limitées.

Snort

Face aux systèmes de détection d'intrusion complets proposés dans les offres commerciales intégrant sondes, signatures, moyens de distribution, interface graphique, etc. ; les solutions open-source ont progressivement trouvé leur place en tirant partie des avantages d'un développement dans le cadre du logiciel libre. Ainsi, l'IDS le plus répandu à l'heure actuelle est probablement le logiciel open-source Snort (http://www.snort.org/) qui est une sonde de détection d'intrusion réseau (NIDS). Le modèle open-source appliqué à Snort a notamment permis un développement plutôt rapide d'une large base de signatures (presque 3000 à ce jour) appuyée essentiellement sur le volontariat et un langage de définition totalement public. Cette base est désormais mise à jour très rapidement et diffusée par Internet (sur le Web par exemple). Par ailleurs, la limitation du projet à la réalisation d'une sonde de détection d'intrusion réseau a permis des améliorations successives très importantes en terme de performance et de capacités d'analyse et a également débouché sur la mise à disposition ultérieure d'extensions orientées vers la diffusion des alertes, le stockage dans les bases de données, etc.

Les signatures Snort font désormais partie du bagage que les administrateurs de sécurité doivent pouvoir utiliser pour envisager la détection d'intrusion et la réaction rapide à de nouvelles attaques (ne serait-ce que pour disposer des détails précis de caractérisation d'un flux d'attaque). A titre d'exemple, nous présentons dans les figures suivantes deux signatures d'attaque (et leurs commentaires) utilisables par Snort respectivement pour détecter une tentative d'exploitation d'une faille grave du service RPC de plusieurs versions de Microsoft Windows et pour repérer le flux généré par un vers baptisé « Klez ».

SID	2251
Message	NETBIOS DCERPC Remote Activation bind attempt
Signature	alert tcp $EXTERNAL_NET any -> $HOME_NET 135 (msg:"NETBIOS DCERPC Remote Activation bind attempt"; content:"|05|"; distance:0; within:1; content:"|0b|"; distance:1; within:1; byte_test:1,&,1,0,relative; content:"|B8 4A 9F 4D 1C 7D CF 11 86 1E 00 20 AF 6E 7C 57|"; distance:29; within:16; reference:cve,CAN-2003-0352; classtype:attempted-admin; reference:url,www.microsoft.com/technet/security/bulletin/MS03-026.asp; reference:cve,CAN-2003-0715; sid:2251; rev:1;)
Summary	This event is generated when an attempt is made to exploit a known vulnerablity in Microsoft RPCSS service for RPC.
Impact	Denial of Service. Possible execution of arbitrary code leading to unauthorized remote administrative access.
Detailed Information	A vulnerability exists in Microsoft RPCSS Service that handles RPC DCOM requests such that execution of arbitrary code or a Denial of Service condition can be issued against a host by sending malformed data via RPC. The Distributed Component Object Model (DCOM) handles DCOM requests sent by clients to a server using RPC. A malformed request to the host running the RPCSS service may result in a buffer overflow condition that will present the attacker with the opportunity to execute arbitrary code with the privileges of the local system account. Alternatively the attacker could also cause the RPC service to stop answering RPC requests and thus cause a Denial of Service condition to occur.
Affected Systems	Windows NT 4.0 Workstation and Server Windows NT 4.0 Terminal Server Edition Windows 2000 Windows XP

Signature Snort pour une attaque MS/RPC

SID	1800
Message	VIRUS Klez Incoming
Signature	alert tcp $EXTERNAL_NET any -> $SMTP_SERVERS 25 (msg:"VIRUS Klez Incoming"; flow:to_server,established; dsize:>120; content:"MIME"; content:"VGhpcyBwcm9"; classtype:misc-activity; sid:1800; rev:3;)
Summary	This event is generated when an incoming email containing the Klez worm is detected.
Impact	System compromise and further infection of target hosts.
Detailed Information	W32/Klez.h@MM exploits the vulnerability in Microsoft Internet Explorer (ver 5.01 or 5.5 without SP2), enabling it to execute email attachments. Once executed, it can unload several processes including Anti-virus programs. The worm is able to propagate over the network by copying itself to network shares (assuming sufficient permissions exist). Target filenames are chosen randomly, and can have single or double file extensions.
Affected Systems	Microsoft Internet Explorer (ver 5.01 or 5.5 without SP2)
Attack Scenarios	This virus can be considered a blended threat. It mass-mails itself to email addresses found on the local system, then exploits a known vulnerability, spreads via network shares, infects executables on the local system.
Ease of Attack	Simple. This is worm activity.
False Positives	Certain binary file email attachments can trigger this alert.
False Negatives	None known.
Corrective Action	Apply the appropriate vendor suppled patches. Block incoming attachments with .bat, .exe, .pif, and .scr extensions
Contributors	Sourcefire Research Team Brian Caswell <bmc@sourcefire.com> Nigel Houghton <nigel.houghton@sourcefire.com> Snort documentation contributed by Nawapong Nakjang (tony@ksc.net, tonie@thai.com)
References	

Signature Snort pour le virus Klez

On voit que ces signatures correspondent à la recherche de séquences particulières dans un flux réseau TCP (sur des ports ou pour des services particuliers). La sonde Snort se charge d'effectuer le ré-assemblage de la séquence TCP parmi tous les paquets IP se présentant devant son interface de capture malgré toutes les techniques de dissimulation éventuellement utilisées (fragmentation, déséquencement, etc.) et de réaliser la recherche des différentes signatures efficacement.

La qualité des signatures disponibles autour de Snort (que ce soit dans la distribution principale du logiciel, ou sur le site Web associé: http://www.snort.org/dl/rules/ et http://www.snort.org/snort-db/) semble globalement assez bonne ; même si elles sont fournies sur la base du volontariat (ce qui ne veut pas dire qu'elles ne génèrent pas quand même un nombre important de fausses alarmes dans un réseau actif...).

Prelude-IDS

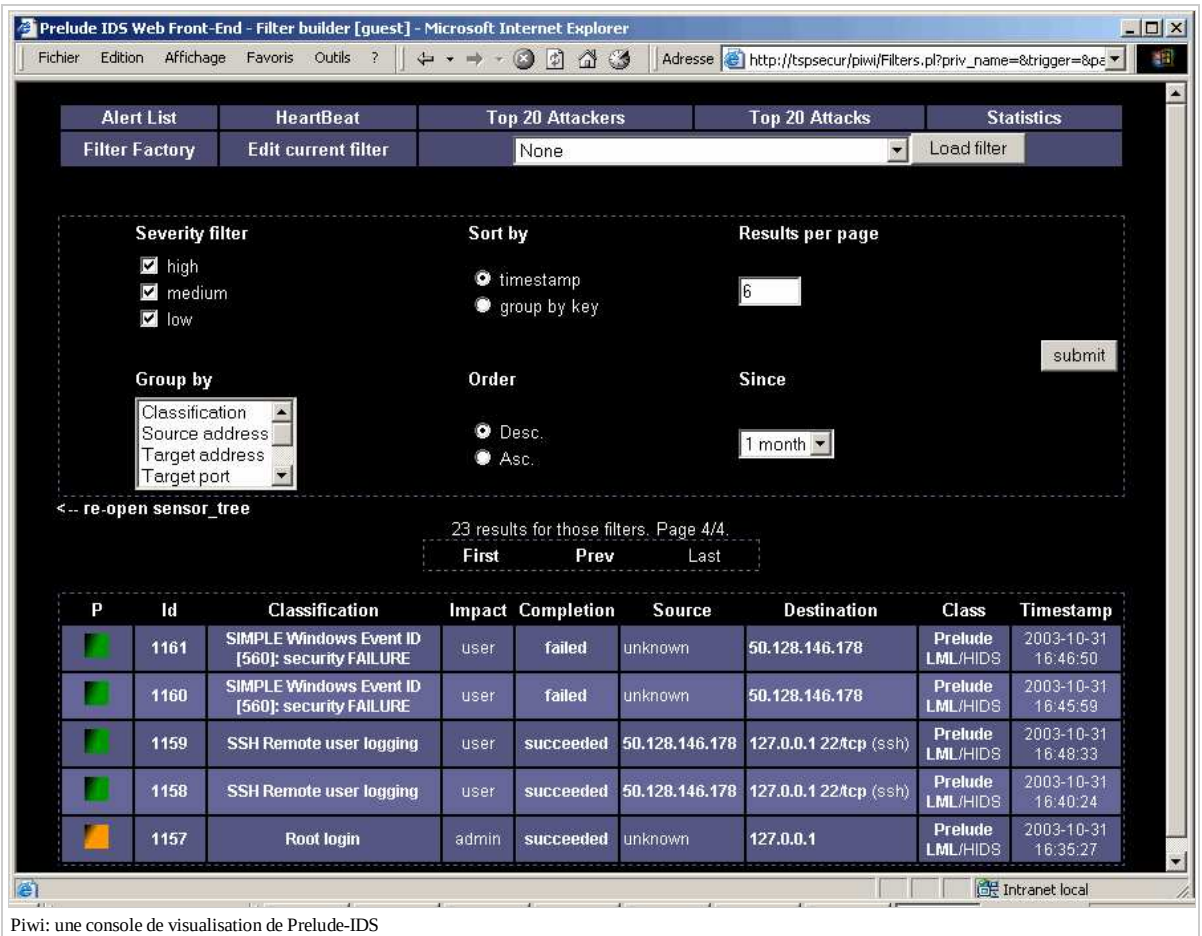
Contrairement au projet Snort, focalisé sur la réalisation d'une sonde spécifique, le projet Prelude-IDS (http://www.prelude-ids.org/) propose un système plus complet comprenant :

- une infrastructure logicielle (librairie, format de communication) permettant de développer différentes sondes intégrées dans l'infrastructure Prelude ;
- un manager capable de collecter les alertes issues de ces différentes sondes et de les stocker dans une base de données (MySQL en général) ;
- un certain nombre de sondes, avec notamment :
  - un NIDS capable de ré-utiliser les signatures de Snort, et d'effectuer d'autres analyses réseau - celui-ci a désormais été déclaré obsolète en faveur de l'utilisation directe de Snort;
  - un HIDS permettant d'analyser différents types de traces (en général au format syslog) ;
  - un certain nombre de patch permettant d'adapter d'autres logiciels de sécurité pour qu'ils émettent des alertes en direction de Prelude, comptant notamment : Snort lui-même nativement, pflog (traces du firewall pf d'OpenBSD), systrace (contrôle des appels système), honeyd (pot de miel) et Nessus (outil de recherche de vulnérabilités) ;
- une interface de visualisation des différentes alertes générées nommée Piwi (en Perl), interface qui est en train d'être remplacée par une autre, nommée Prewikka, en cours de développement pour permettre un contrôle plus complet des autres composants.

D'après notre expérience, l'ensemble logiciel Prelude-IDS (v.0.8 et v.0.9, c'est à dire les deux dernières versions stables) est parfaitement capable de gérer correctement un système de détection d'intrusion composé de 2 ou 3 sondes déployés sur un système informatique réel de bonne taille en production depuis prés d'un an. Les mécanismes de collecte d'alertes et de communication nous semblent donc tout à fait stables, tout comme le stockage des alertes dans une base de données externe.

Pour l'instant, les possibilités de contrôle du système de manière centralisée depuis l'un des manager sont limitées, ces aspects doivent être pris en compte par des moyens d'administration conventionnels, mais comme les différents composants (notamment les sondes) gèrent correctement les arrêt/relance du point de vue de la communication avec le manager ceci n'est pas bloquant (avec un nombre limité de sondes). Ce point devrait faire l'objet de certaines des nouvelles fonctionnalités dans la prochaine version du système de manière à permettre un contrôle centralisé des sondes (et notamment de leur configuration de détection).

L'interface d'accès aux alertes de Prelude est encore assez limitée. La version précédente, Piwi, est limitée à des possibilités de visualisation simples via un navigateur Web et un serveur HTTP (via des cgi Perl). La figure suivante présente un exemple de cette interface. Piwi permet aussi de réaliser un certain nombre de tris et de filtrages sur les requêtes d'interrogation SQL sous-jacentes, mais tout travail d'analyse plus poussé demande d'interroger directement la base de données. Il n'est pour l'instant pas possible d'agir sur le contenu de la base de données (ce qui serait, à notre sens, le premier pas vers des fonctions de corrélation et d'agrégation d'alertes). À nouveau, une autre version de l'interface est en cours de développement, sur des bases totalement différentes.



Piwi: une console de visualisation de Prelude-IDS

Globalement, Prelude-IDS est un logiciel qui nous semble particulièrement prometteur, du fait qu'il jette les bases d'un système de détection d'intrusion complet, déjà capable de remplir les fonctions d'une sonde réseau ou système efficace, mais également susceptible de rassembler des informations provenant de plusieurs sondes de différents types dans un même entrepôt de données. Sur cette base, l'intégration de fonctions de groupement et de corrélation avancées nous semble même alors possible.

Traitement des alertes

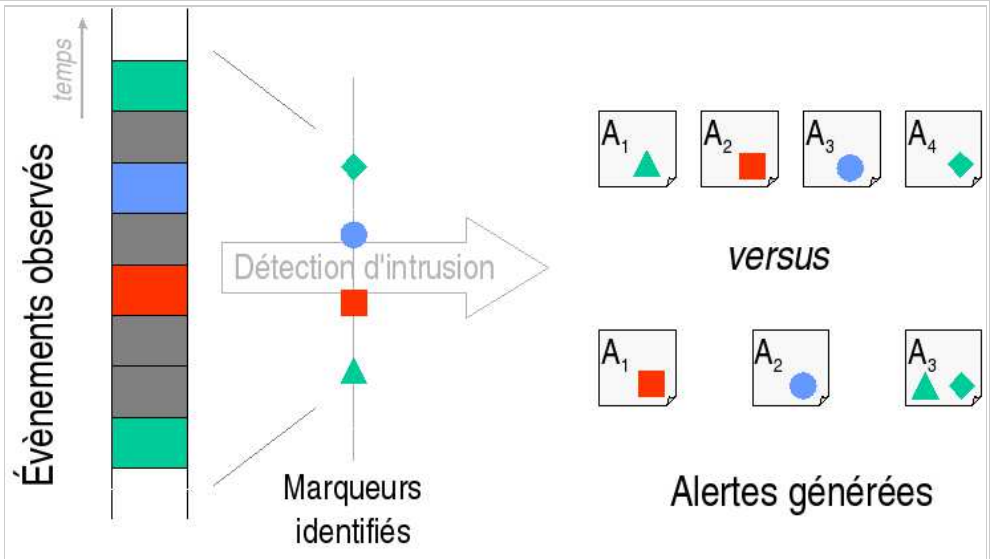
Limites

Notamment dans le domaine des sondes de détection d'intrusion réseau, les techniques de détection utilisant une application de signatures sur les évènements observés vont avoir pour effet de sélectionner certains des évènements, contenant un marqueur spécifique à l'origine de l'activation d'une signature. La plupart du

temps, chacun des évènements contenant ces marqueurs va être à l'origine de la génération d'une alerte. Pourtant, en général, il serait souhaitable que les alertes soient d'un niveau d'abstraction plus élevé, et qu'elles rassemblent différents marqueurs correspondant à des évènements différents mais associés à une même action.

Dans cette vision, il ne s'agirait pas véritablement de demander à l'IDS d'établir des liaisons de corrélations (cause à effet, topologie, etc.) mais plutôt d'agréger correctement les alertes générées de manière à garantir la correspondance entre chaque alerte et au plus une action. Toutefois, ce besoin correspond à la mise en oeuvre d'une analyse multi-évènements : une même action pouvant être à l'origine de plusieurs évènements (par exemple, le déclenchement d'une attaque réseau peut conduire à la génération de plusieurs paquets IP), la sonde devrait être capable de rechercher des similarités entre marqueurs appartenant à plusieurs d'entre eux. Elle devrait donc pouvoir réaliser une analyse multi-évènements.

Dans la représentation de ce problème que nous donnons dans la figure suivante, la forme des marqueurs est ce qui permet à la sonde de les repérer mais, suivant que la couleur des évènements auxquels ils appartiennent est prise également en compte ou non, le nombre d'alertes générées est différent. Ainsi, la génération de l'alerte *A4* pourrait être évitée.



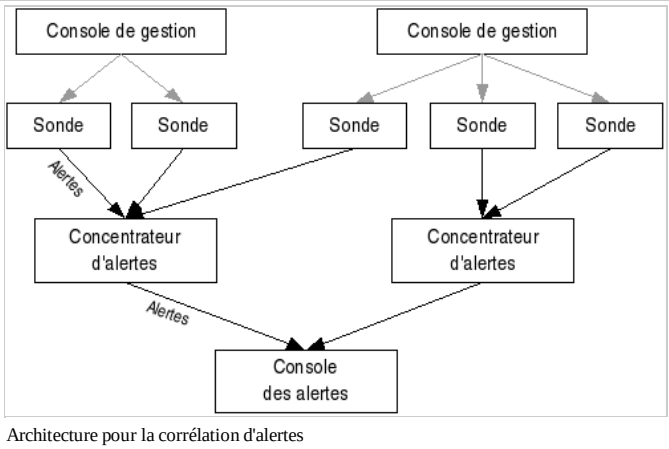
Analyse multi-évènements

Toutefois, on voit bien sur cette illustration que l'analyse multi-évènements exige de prendre en compte les relations existant entre deux évènements successifs, parfois séparés dans le temps par un grand nombre d'autres évènements, et même d'autres alertes. Le travail d'analyse de la sonde est donc largement plus difficile. Par ailleurs, l'alerte *A3* dans le cas multi-évènements est générée tardivement par rapport à l'alerte *A1* dans le cas mono-évènement. Si l'agrégation de plusieurs marqueurs facilite le diagnostic sécurité, d'un autre côté, ce retard peut aussi être préjudiciable à la sécurité. Ce compromis de mise en oeuvre complique également la conception de la sonde.

Pour aborder ce type de traitement, que ce soit au niveau des sondes elles-mêmes ou au niveau des systèmes de collecte des alertes, il nous semble que les IDS actuels devraient intégrer des fonctions nouvelles, généralement regroupées sous la désignation de corrélation d'alertes.

Architecture de corrélation d'alertes

L'architecture IDWG n'est pas totalement adaptée pour couvrir complètement les besoins de la corrélation d'alertes. Une architecture mieux adaptée est présentée dans la figure suivante, qui dissocie les consoles de gestion des différentes sondes et la console des alertes qui peut bénéficier des traitement d'autres outils de concentration d'alertes.



Architecture pour la corrélation d'alertes

Dans cette architecture, les concentrateurs d'alertes peuvent jouer un rôle vis à vis de la collecte et du transport, mais ils peuvent également traiter les alertes pour réaliser des groupements ou certaines corrélations et fournir au niveau de la console des alertes une information agrégée (à la place ou en complément des alertes originelles). Les fonctions de corrélation les plus avancées ou nécessitant la connaissance de l'ensemble des alertes générées dans le système, doivent elles être réalisées au niveau de cette console. (La console n'est pas nécessairement associée directement à l'interface homme-machine de l'IDS.) Les associations entre les différentes sondes et les concentrateurs d'alertes sont établies en fonction de ces possibilités de corrélation, et elles peuvent être différentes des associations nécessaires pour la gestion technique des sondes réalisée par les consoles de gestion. Les secondes peuvent être imposées par les constructeurs par exemple, tandis que les premières peuvent s'appuyer sur la nature des sondes (par exemple, NIDS d'une part, HIDS d'autre part).

Groupement

Parmi les fonctions de corrélation envisageables, la première que nous appellerons plutôt une fonction de groupement consiste à rassembler dans une même alerte (virtuelle) des alertes similaires provenant d'un même événement.

Ces groupements peuvent permettre, par exemple, de regrouper des alertes émises par des sondes différentes observant plusieurs fois le même événement (par exemple le long de son trajet pour des sondes réseau). Ce type de regroupement n'est pas si facile à réaliser : dans le cas de sondes utilisant des bases de connaissance différentes, les nomenclatures des attaques peuvent être très différentes et nécessiter la mise au point de tables de correspondance importantes. De tels regroupements sont également plus intéressants qu'il n'y paraît : outre qu'ils diminuent le nombre d'alertes à traiter pour l'opérateur, ils peuvent permettre d'enrichir la description de l'alerte virtuelle globale. En effet, des sondes différentes ont pu renseigner des informations différentes dans les indications de leurs alertes (adresse IP pour l'une, nom de compte pour l'autre par exemple).

Les autres types de regroupement les plus immédiats consistent à rassembler certaines attaques concernant une même source ou une même destination. Ainsi, un scan (une recherche active de vulnérabilités) réalisé par un outil donné (ou certains virus) donnera probablement lieu à de nombreuses alertes qui peuvent être regroupées avec une bonne fiabilité si elles suivent une séquence spécifique et ciblent la même destination. C'est également le cas si des alertes similaires provenant de la même source visent des destinations successives. Dans un cas comme dans l'autre, le regroupement est certainement à faire avec une certaine prudence, mais le gain en terme de lisibilité pour l'alerte virtuelle globale par rapport à l'ensemble des alertes élémentaires est très important, tout comme le gain de temps pour les opérateurs.

Enfin, une autre catégorie de regroupement qui semble souhaitable, et qui est liée à la précédente, c'est bien entendu le regroupement temporel. Des alertes périodiques, ou des alertes ayant lieu dans un intervalle de temps réduit peuvent donner lieu à un premier regroupement, ce qui est d'autant plus intéressant si elles présentent des similarités du type évoqué précédemment. Toutefois, les intervalles de temps à utiliser sont difficiles à évaluer ; surtout si on prend en compte le fait qu'un attaquant intelligent puisse prévoir ce type de regroupement et tenter de l'éviter ou de le détourner. La notion de proximité temporelle, pour des alertes de sécurité, reste à manier avec précaution. Mais c'est une information très importante qu'il est impossible d'ignorer quand on réalise un traitement des alertes générées par les IDS. Un effort d'automatisation sur ce point nous semble vraiment souhaitable.

Corrélation

Outre le regroupement des alertes en alertes virtuelles plus faciles à gérer pour les administrateurs, on pourrait également attendre d'un IDS qu'il soit en mesure de réaliser un suivi de séquences d'alertes afin d'essayer de fournir un diagnostic plus complet d'une intrusion éventuelle ou d'éliminer des fausses alarmes.

Ce raisonnement part de l'hypothèse qu'en cas d'intrusion, les actions d'attaque effectuées par un intrus suivront une certaine logique d'enchaînement et que des attaques successives (et donc des alertes successives) présenteront des liens entre elles permettant à la fois de confirmer l'alarme suggérée par chaque alerte prise isolément, mais également d'établir un diagnostic plus complet de l'intrusion, de ses objectifs, de son niveau d'avancement, etc. ; et donc du danger associé.

Pour réaliser de telles corrélations, un IDS devrait disposer d'une base de connaissance décrivant non seulement les attaques élémentaires, mais également les enchaînements possibles entre différentes attaques élémentaires pouvant constituer un scénario d'intrusion plus élaboré. Il s'agit schématiquement de décrire les liens de cause à effet entre attaques afin de permettre à l'IDS de raisonner sur les liens existant entre les alertes qu'il connaît (ou plus précisément les attaques auxquelles elles correspondent). Dans certains cas, on peut même envisager que l'IDS soit en mesure d'effectuer des hypothèses sur l'existence de certaines actions non-observées (ou inobservables) afin de compléter des scénarios d'intrusion. Enfin, en établissant le lien entre les préconditions des attaques et les vulnérabilités connues recensées dans le système, ou entre les effets de ces attaques et les objectifs de sécurité du système, l'identification des possibilités de succès de l'intrusion et de son niveau de danger dans le système visé pourrait également être envisagée. De telles fonctions permettraient bien évidemment de faciliter énormément le travail de l'administrateur de sécurité chargé de superviser l'IDS auquel serait fourni une proposition de diagnostic et un niveau de risque plus réfléchis, basés sur les évènements observés par les différentes sondes du système.

L'approche de la corrélation que nous présentons ici est avant tout basée sur une vision logique. Le fonctionnement du moteur de corrélation et la structure du diagnostic sont basés sur des raisonnements logiques concernant les attaques, leur faisabilité, les possibilités d'enchaînement, etc. D'autres approches ont été proposées, appuyées sur des techniques statistiques de caractérisation ou de classification des alertes. Dans tous les cas, ces travaux n'ont pour l'instant pas réellement vus de mise en œuvre effective dans des implémentations opérationnelles de système de détection d'intrusion. Par contre, un certain nombre d'efforts ont déjà était faits dans certaines implémentations pour faciliter le rapprochement entre diverses sources d'information, notamment en vue de limiter le nombre de fausses alertes.

Sécurité informatique/Centralisation des traces

Au travers de la détection d'intrusion, on voit que la collecte d'information dans le système informatique est une opération importante pour traiter ses problèmes de sécurité. Dans certains cas, la centralisation de cette information est quasiment vue comme une fin en soi, et associée à une propriété de la sécurité qui a été baptisée l'auditabilité. De notre point de vue, cette propriété n'est pas réellement dissociée des autres propriétés de sécurité ; par contre, elle met en relief un certain nombre de besoins de sécurité qui impliquent généralement la collecte et le stockage (souvent de manière centralisée) des sources d'information disponibles dans le système informatique. Nous regroupons ces opérations sous le thème de la centralisation des traces dans le système informatique.

Le problème de la gestion des traces

Le besoin de collecte peut avoir une origine technique, comme dans les cas où ces traces sont nécessaires pour la mise en œuvre d'un système de détection d'intrusion ou la réalisation de contrôles liés à la sécurité. Mais il peut également avoir une origine plus politique, notamment quand la disponibilité de ces données est liée à l'obligation de pouvoir fournir des éléments d'enquête à des organismes habilités, internes (audit interne) ou externes (tutelles, cour des comptes pour les organismes publics, etc.), voire tout simplement une origine légale pour répondre aux besoins de l'autorité judiciaire. Dans l'ensemble de ces cas, la mise à disposition des traces fournies en standard par les systèmes d'exploitation, les progiciels, les équipements réseau et bien sûr les équipements de sécurité eux-mêmes devrait pouvoir être possible.

Toutefois, pour être réalisable, cette mise à disposition doit réellement avoir été prévue. Par ailleurs, pour être un tant soit peu probantes, les traces collectées doivent présenter un minimum de garanties d'intégrité - la deuxième action d'un attaquant averti étant d'effacer les traces qu'il génère.

Enfin, il ne faut pas oublier que, à partir du moment où ces traces contiennent des informations nominatives - il est quasiment inévitable qu'elles puissent en contenir si elles sont dignes d'intérêt - elles doivent être gérées en conformité avec les directives de la CNIL et de la loi « Informatique et libertés » (c'est à dire qu'elles ne doivent pas être conservées indéfiniment et qu'elles doivent être protégées pour qu'on ne les détourne pas de leur finalité originelle). On consultera notamment sur ces points le rapport de la CNIL sur « La cybersurveillance sur les lieux de travail », disponible à l'adresse : http://www.cnil.fr/fileadmin/documents /approfondir/rapports/Rcybersurveillance-2004-VD.pdf.

A l'heure actuelle, il nous semble que ces besoins peuvent difficilement être satisfaits autrement que par la mise en place pragmatique d'un système centralisé de consolidation et de stockage des traces, géré directement par les acteurs de la SSI. Il semble même que syslog soit un standard de fait incontournable pour prendre en compte une bonne majorité des types d'équipement existants (même si cela demande des adaptations pour les systèmes basés sur MS/Windows).

Mise en oeuvre

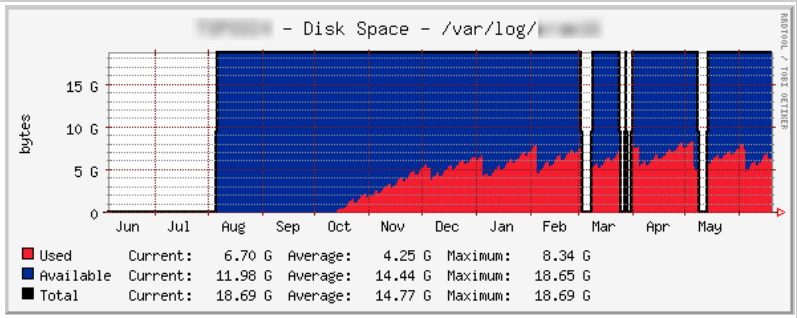
La solution la plus directe: Syslog

Dans la pratique, nous recommanderions tout simplement l'utilisation d'un serveur Unix `syslog-ng` (pour faciliter le tri des traces en fonction de leur origine) détaché du reste du système informatique et décemment configuré du point de vue sécurité, équipé d'une zone de stockage relativement grande et assurant une rotation régulière des traces collectées, par exemple avec l'utilitaire logrotate. (Un système Debian GNU/Linux standard assure ces fonctions sans difficultés dans la configuration par défaut.) Les systèmes incapables d'utiliser Syslog ne sont généralement pas capables de faire plus que de stocker leurs traces dans des fichiers à plat, lesquels dans ce cas doivent être déportés manuellement s'ils sont intéressants, par exemple via SSH. Ce type de solution est loin d'être idéal du point de vue de sa propre sécurité, surtout si on le compare, par exemple, avec les modes de transmission d'alertes de certains IDS (via des connexions SSL authentifiées avec des certificats). Syslog est un protocole fonctionnant sur UDP [RFC 3164] ou éventuellement sur TCP [RFC 3185], notoirement non-sûr. Toutefois, dans la pratique, l'intérêt concret d'un tel système (s'il est utilisé par les autres éléments du système informatique pour stocker leurs traces) est très important, à la fois en terme de données disponibles pour la surveillance de la sécurité et pour pouvoir disposer d'informations en cas d'intrusion grave. Et la protection d'une telle machine est relativement facile à réaliser.

Eléments de volumétrie

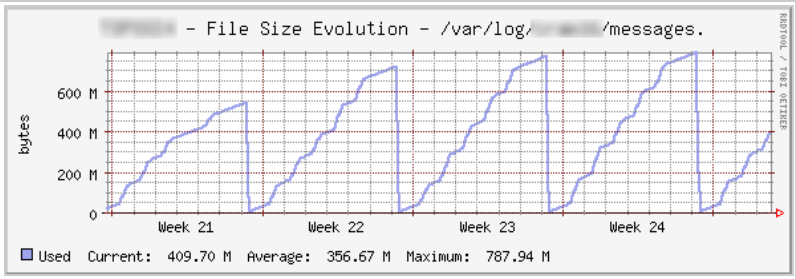
Nous mentionnons dans cette section quelques données quantitatives relatives à l'utilisation d'un serveur Syslog de centralisation. Les deux figures suivantes permettent de donner une idée (empiriquement) des capacités de stockages (et d'analyse) nécessaires pour des systèmes courants. Toutefois, ces données correspondent bien sûr à des cas particuliers. Suivant les applications concernées, les volumes peuvent être trés différents. Ces éléments correspondent à un système informatique comptant environ un millier d'utilisateurs et autant de postes de travail sur lequel les données de sécurité courantes disponibles avec le système d'authentification basé sur Active Directory sont collectées, centralisées, conservées pendant une durée définie puis *détruites* automatiquement (afin de respecter à la fois les besoins de sécurité des données et des utilisateurs).

Au cours du processus de rotation, les traces sont compressées de manière à maitriser l'espace disque occupé. Les transferts se font au fil de l'eau via UDP, dans le mode de fonctionnement nominal de Syslog. La configuration des paramètres d'émission des traces (niveau d'audit, stratégie d'audit, *debug* ou *logging* level, etc.) peut avoir un effet trés important sur les volumes impliqués (ainsi que sur la pertinence des traces générées).



Exemple d'évolution de l'occupation disque

La première figure ci-dessus montre l'évolution de l'occupation de la partition de stockage des différentes traces sur une longue période.



Evolution de la taille d'un fichier de traces A.D.

La seconde figure détaille plus particulièrement l'évolution de la taille du fichier plat dans lequel l'ensemble des traces sont enregistrées sur une période plus courte. Ce fichier là est non-compressé et subit une rotation hebdomadaire.

Sécurité informatique/Observation, surveillance, supervision

Les systèmes de surveillance dédiés à la sécurité, comme les IDS notamment, partagent avec certaines solutions d'administration réseau un certain nombre de caractéristiques techniques. Notamment, les solutions dédiées à l'observation du réseau (pour le dépannage en particulier), la surveillance de l'exploitation, et la supervision des systèmes partagent avec les systèmes de sécurité des sources d'information et des modalités de mise en place communes. De notre point de vue, ces différentes désignations concernent en général des solutions différentes (quoique assez proches) que nous définissons dans ce texte de la manière suivante :

- observation (réseau) : les moyens logiciels et matériels dédiées à l'observation réseau permettent de prélever et reconstituer un flux réseau à des fins d'analyse. On trouve dans cette catégorie des équipements matériels permettant d'intercepter le flux réseau notamment dans les réseaux full duplex commutés, et beaucoup de logiciels dont les capacités vont de la capture pure et simple de paquets IP à la possibilité de reconstituer les différents flux de transmission au niveau de nombreuses applications réseau.
- surveillance : les moyens de surveillance sont habituellement orientés vers la surveillance régulière du bon fonctionnement des applications importantes et des services essentiels au bon fonctionnement du système informatique (DNS par exemple) et l'émission d'alertes en cas de dysfonctionnements. Ces systèmes peuvent également être destinataires des alertes générées par les équipements eux-mêmes quand ils en sont capables (par exemple via l'utilisation de traps SNMP pour les équipements supportant ce type de protocole de gestion). Ces outils visent à améliorer le fonctionnement du système, en détectant rapidement des défaillances pour y pallier efficacement.
- supervision : enfin, les moyens de supervision, qui peuvent également contacter régulièrement un certain nombre d'autres équipements pour obtenir des statistiques d'utilisation (et parfois partager certaines informations avec les outils de surveillance précédents) sont plus orientés vers la collecte régulière de mesures de fonctionnement et la constitution de données consolidées (par exemple pour les latences réseaux, le débit réseau, le temps de réaction d'une application, l'occupation disque, etc.). Ces outils sont particulièrement utiles pour évaluer le dimensionnement des plate-formes matérielles en fonction de l'historique de l'utilisation ou pour identifier des sources de dégradation des performances.

Quoique ces solutions ne soient pas directement liées à la sécurité des systèmes informatiques, au sens de la protection contre les malveillances, leur utilisation participe d'une bonne maîtrise du système dans son ensemble et, parmi les outils utilisables, certains font partie de la boite à outils de l'administrateur sécurité. Une bonne appréciation du fonctionnement du système informatique fournit des pistes pour appréhender le système dans son ensemble : les flux réseaux principaux, les services clefs, le détail des transmissions d'une application par exemple. À certains moments, ces informations complètent celles fournies par les outils dédiés à la sécurité. Par ailleurs, les outils concernés peuvent également fournir des informations concernant la sécurité directement, à condition parfois de s'intéresser à des paramètres habituellement peu observés (comme les taux d'erreurs par exemple).

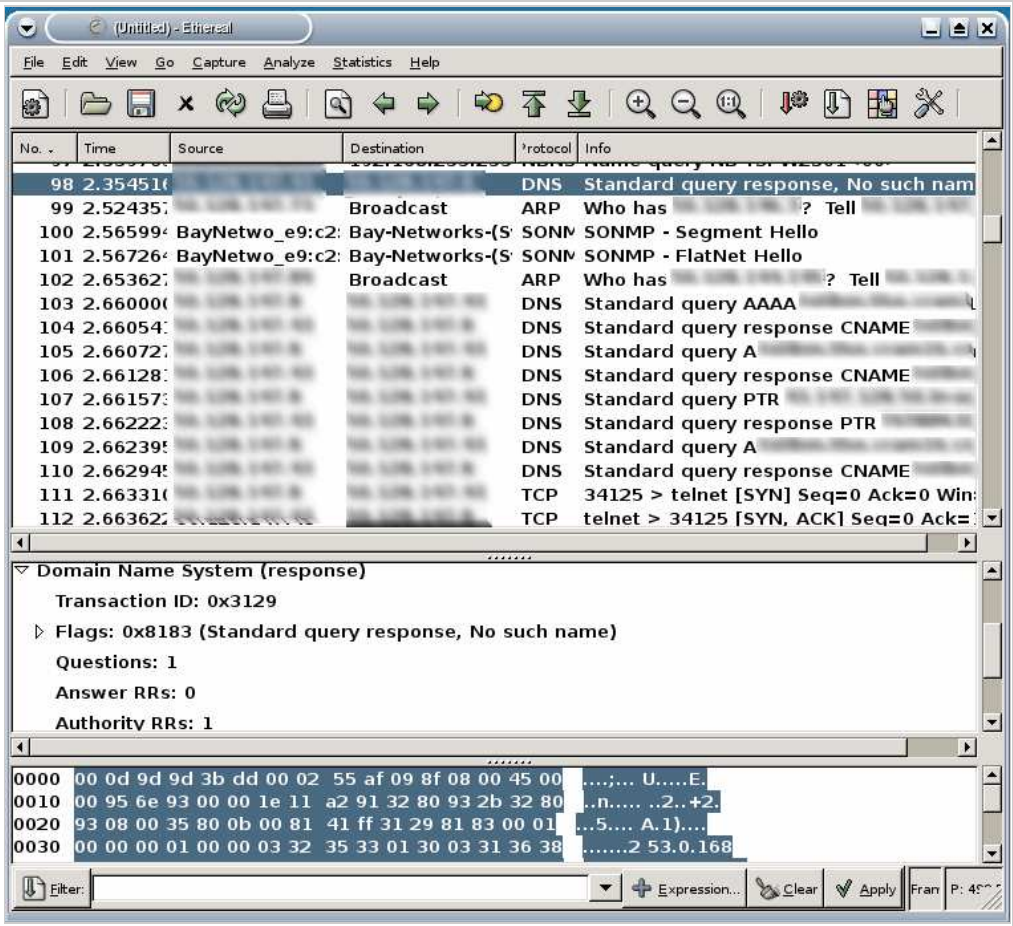
Afin d'illustrer ces liens, nous nous appuyons dans cette section sur certaines des solutions open-source disponibles pour aborder les problématiques d'observation réseau, de surveillance des systèmes ou de supervision. L'ouverture et la disponibilité de ces logiciels permet de bien comprendre le mode de fonctionnement de ces différents outils et les paramètres auxquels ils donnent accès. En ce qui concerne l'observation réseau, nous considérerons Ethereal, un outil de capture et d'analyse des flux réseau particulièrement efficace et de plus en plus répandu, en ce qui concerne la surveillance, nous parlerons de Nagios, outil de vérification du fonctionnement des services et d'alerte, enfin, en ce qui concerne la supervision, nous parlerons de Cacti et de rrdtool, outils permettant de collecter et de consolider des données issues des capteurs et des compteurs mis à dispositions par les systèmes informatiques (et si possible de ntop, un outil de suivi général du réseau).

Les outils de supervision et de surveillance ont besoin d'un moyen le plus général possible pour accéder aux informations internes des différents systèmes informatiques qu'ils observent, si possible de manière indépendante des constructeurs ou des systèmes d'exploitation. Souvent, le support limité du constructeur en terme de pilotes ou de logiciels implique d'utiliser des solutions propriétaires. Toutefois, le protocole SNMP (Simple Network Management Protocol) défini dans le standard IETF STD 62 (ou les RFC 3411-3418) est un standard suffisamment répandu et suffisamment puissant pour permettre de couvrir une large gamme de systèmes et de données. Ce n'est malheureusement pas une solution à tous les problèmes, mais dans les exemples que nous prendrons, SNMP sera souvent la technologie la plus simple utilisée de manière sous-jacente pour collecter des données de supervision ou de surveillance.

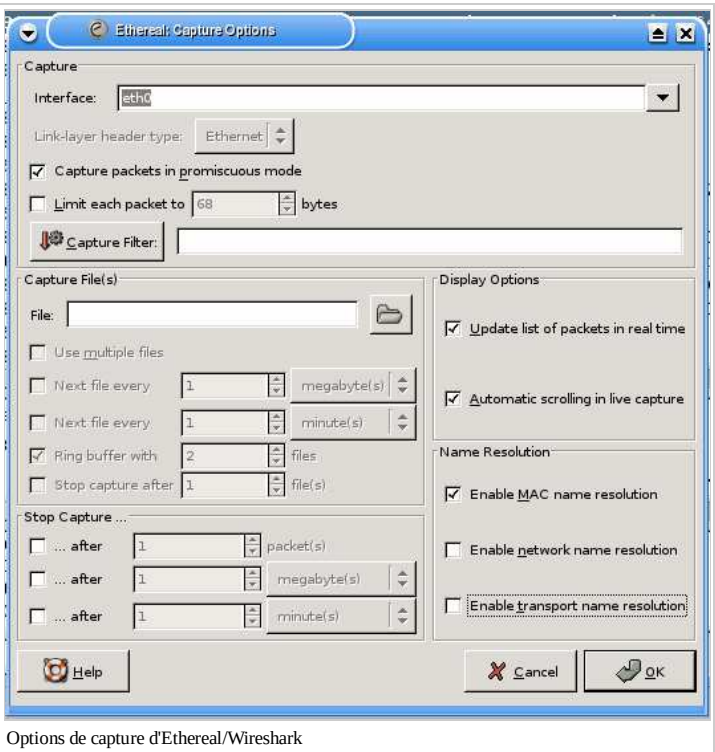
Ethereal/Wireshark (http://www.wireshark.org/)

Ethereal (trés récemment renommé Wireshark (http://www.wireshark.org/)) est un outil d'analyse réseau qui permet à la fois de réaliser une capture du trafic visible depuis une interface réseau de la machine sur laquelle il s'exécute, et de visualiser plus précisément le contenu des paquets capturés en fonction du protocole réseau auquel ils correspondent. Ethereal est désormais capable d'analyser la plupart des protocoles réseaux existants. Il est de ce fait devenu un outil incontournable dans la boite à outil de l'administrateur sécurité, notamment parce qu'il permet d'étudier en détail le contenu d'une capture réseau (obtenue éventuellement par des moyens différents) et parce qu'il permet d'identifier précisément les flux réseaux associés à une application (en observant le trafic réseau associé à son exécution) et donc de mieux comprendre son fonctionnement. Ce dernier point peut être particulièrement utile pour mettre en place un filtrage réseau pour l'application concernée au niveau d'un firewall. L'outil est disponible pour plusieurs version d'Unix, mais également pour MS/Windows.

Les figures suivantes présentent un exemple de capture de trafic réalisée avec Ethereal et le détail des options de capture disponibles.

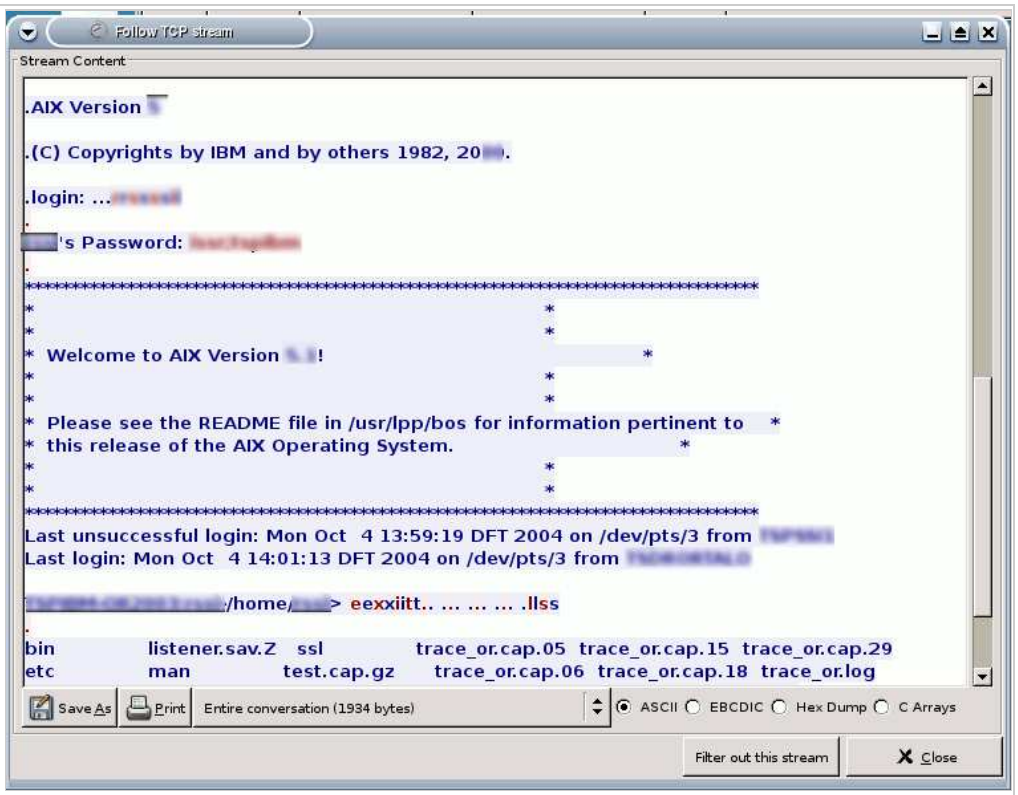


L'outil d'observation réseau Ethereal/Wireshark



Options de capture d'Ethereal/Wireshark

La figure suivante montre une analyse plus détaillée d'une connexion TCP capturée dans le flux général. S'agissant d'une session Telnet, on y voit d'ailleurs au passage un exemple concret de la grande vulnérabilité de ce protocole et de son mode d'authentification en cas de capture réseau.

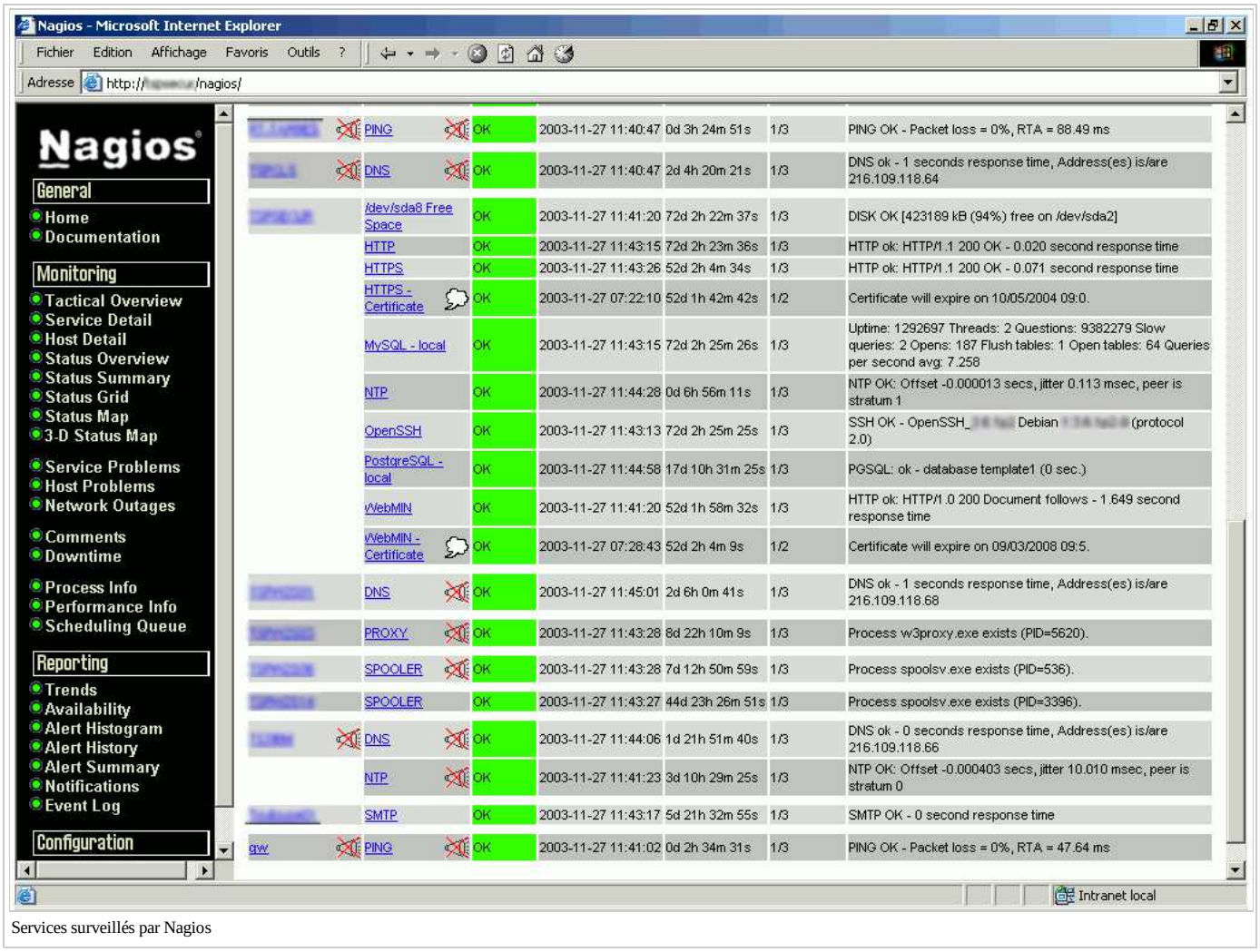


Reconstruction d'une session Telnet avec Ethereal/Wireshark

Ethereal est un outil appuyé sur une interface graphique, mais on dispose aussi d'une version fonctionnant purement dans un terminal texte : tethereal. Cette version permet aussi de réaliser des captures réseau directes en ligne de commande et peut se substituer avec bénéfice à l'outil classiquement utilisé sous Unix pour ce type d'opération : tcpdump.

Nagios

Nagios (http://www.nagios.org/) est un outil offrant la possibilité d'exécuter de manière périodique des scripts ou des programmes de vérification du bon fonctionnement d'un service (généralement exécuté sur une machine distante). Nagios surveille par ce moyen l'état du service considéré, qui peut être classé parmi les catégories OK, WARNING, CRITICAL ou UNKNOWN. Nagios met à jour périodiquement l'état des services qu'il doit surveiller, en prenant en compte les problèmes survenant seulement de manière transitoire, les problèmes liés à la plate-forme, au réseau, et en enregistrant les périodes d'inactivité d'un service (notamment pour réaliser des rapports sur sa disponibilité).



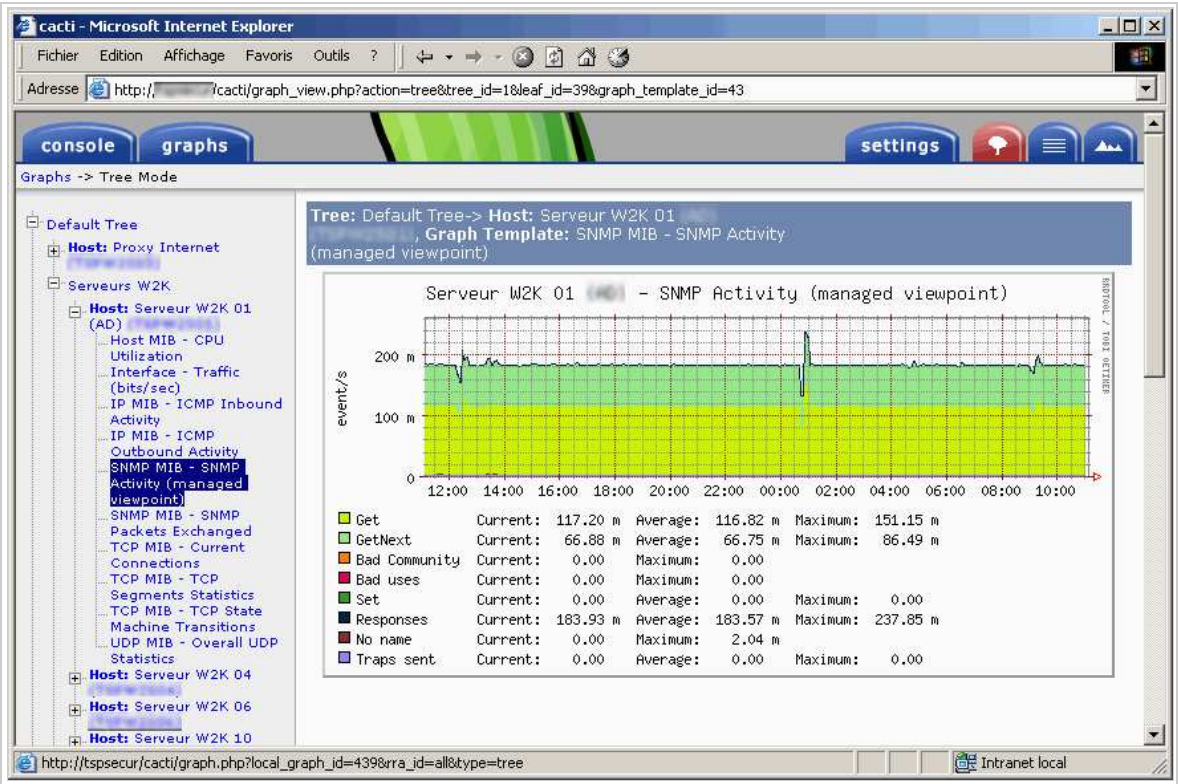
Services surveillés par Nagios

La configuration de Nagios s'effectue par l'intermédiaire de fichiers texte, mais le langage de configuration permet la définition de modèles (pour réutiliser des définitions) et l'outil est également doublé d'une interface via un serveur HTTP et plusieurs programmes « cgi ». Cette interface permet de visualiser l'état des différents services et des différentes machines surveillées et de lancer certaines opérations : lancement ou arrêt de la surveillance (par service), activation ou désactivation de l'émission d'alertes en cas de changement d'état, inscription de périodes de maintenance pour une machine, acquittement d'une alerte (pour un problème en cours de traitement), etc. La figure 52 présente un exemple de cette interface de visualisation. L'intérêt de Nagios dans une configuration particulière dépend bien évidemment de la disponibilité ou non de scripts de surveillance adaptés. Nagios est accompagné d'un certain nombre de plugins pour les actions de surveillance courantes (notamment pour des systèmes Unix). Mais ceux-ci sont parfois insuffisants, notamment dans le cas des systèmes MS/Windows. Dans ce cas, il est toutefois possible de les compléter par des scripts assez simples, à condition de pouvoir s'appuyer sur des requêtes SNMP pour obtenir les informations nécessaires de la machine MS/Windows (état des disques, des périphériques, etc.).

Cacti

Cacti (http://www.cacti.net/) est une application Web écrite en PHP et appuyée sur une base de données (généralement MySQL). Ce logiciel permet de collecter des données (et notamment des données disponibles via SNMP), de les stocker dans une base de données gérées par l'outil RRDTool (voir ci-après) et de piloter les fonctions de création de graphiques disponibles via RRDTool pour afficher sur un navigateur Web une présentation graphique de la consolidation des données collectées.

Cet outil, dont un exemple de résultat est présenté dans la figure suivante, est de la famille des outils de supervision réseau lancée par le très populaire MRTG. L'auteur de MRTG n'est autre que l'auteur de RRDTool vis à vis duquel Cacti se comporte en fait comme une sorte d'interface homme-machine. L'intérêt de Cacti est de rendre beaucoup plus conviviale et parfois extrêmement facile d'utilisation un outil en ligne de commande qui, bien que très puissant, peut parfois être difficile à gérer manuellement quand on gère un nombre important de sources de données et quand on souhaite travailler facilement sur plusieurs présentations.

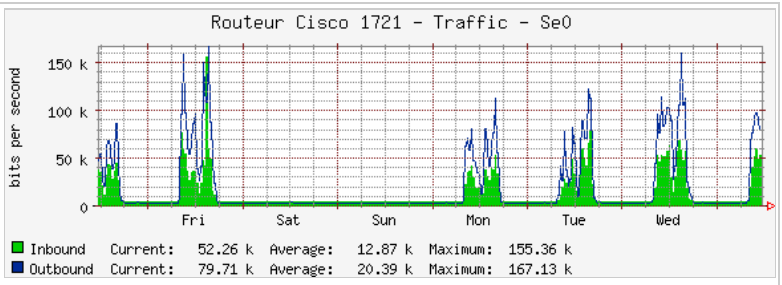


Supervision réseau avec Cacti et SNMP

La figure précédente est un exemple de graphique sophistiqué que l'on peut construire assez facilement dans Cacti depuis la définition des sources de données, jusqu'aux détails de la présentation de la courbe. Il s'agit ici, sur une journée, de données disponibles via SNMP et concernant justement le nombre et le type des requêtes SNMP traitées par un équipement mettent en œuvre ce protocole. L'exemple choisi ne montre aucune activité anormale et seulement la récupération régulière des information de supervision elles-mêmes. Mais si un scan SNMP était effectué sur la machine concernée sans précaution, celui-ci apparaîtrait de manière évidente sur la courbe d'historique.

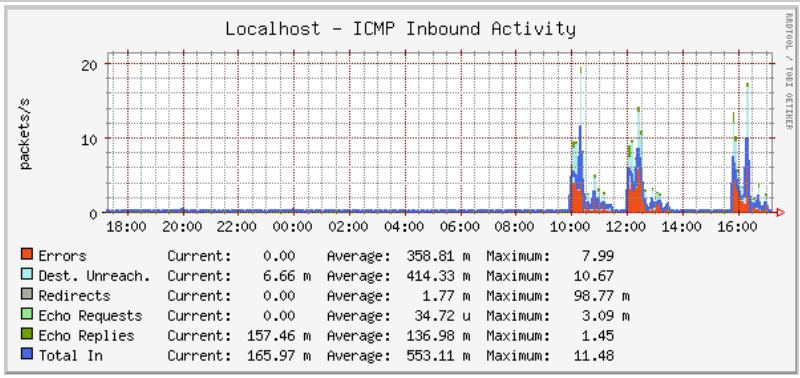
RRDTool

RRDTool (http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/) est l'outil sous-jacent à Cacti pour le stockage et l'affichage graphique des données numériques collectées. La spécificité de RRDTool est de stocker les données qui lui sont fournies non pas sous forme linéaire exhaustive, mais sous une forme immédiatement consolidée. Ainsi, si on enregistre par exemple toutes les 5 minutes les compteurs concernant le nombre de paquets IP émis et reçus par une interface réseau dans une base de données RRDTool, l'outil conservera les données de manière exhaustive sur une journée seulement, et maintiendra en parallèle un certain nombre de données déjà agrégées (par des fonctions de consolidation comme la moyenne, ou le maximum) avec une résolution adaptée à la visualisation souhaitée (en général, pour une semaine avec des valeurs consolidées sur 30 minutes, pour un mois avec des valeurs consolidées sur 2 heures, et pour un an avec des valeurs consolidées sur toute une journée). En effet, il est inutile de conserver la résolution maximale (5 minutes) pour visualiser les données concernant une année sur un graphique qui ne pourrait pas afficher l'ensemble des points de manière lisible. De plus, la consolidation permet de garantir la taille occupée par le fichier contenant les données dès sa création, ce qui évite de remplir progressivement tout l'espace de stockage et de voir exploser le temps de traitement avec une approche naïve consistant à conserver toutes les mesures effectuées. La figure suivante montre un exemple des données des compteurs d'une interface réseau consolidée sur une semaine. C'est un exemple caractéristique de graphe de suivi du trafic réseau avec ce type d'application.



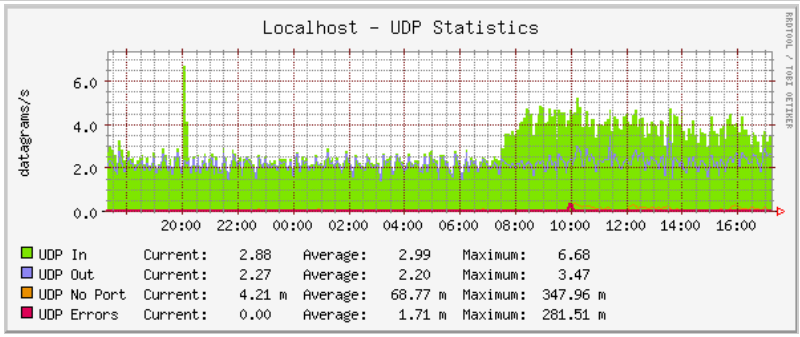
Evolution du trafic réseau

Les données consolidées dans un outil comme RRDTool à des fins de présentation graphique et de supervision, sont parfois révélatrices d'anomalies qui peuvent facilement être rapprochées d'évènements de sécurité. Par exemple, la figure suivante montre le suivi de l'activité ICMP entrante d'une machine ayant effectué un scan de ports vers d'autres machines du réseau. On y voit une augmentation forte du nombre de paquets ICMP destination unreachable et errors entre 10h et 17h : ceux-ci correspondent aux messages d'erreurs envoyés par les machines subissant le scan quand on a tenté de les contacter sur des ports réseau fermés. La figure d'aprés, qui montre l'évolution des statistiques relatives au protocole UDP sur la même période révèle également de légères traces du scan au travers des réponses UDP no ports.



Suivi de l'activité SNMP

Il faut aussi noter qu'un scan délibérément lent pour échapper à une détection pourrait malgré tout être visible sur ce type de courbe (notamment dans les historiques plus longs couvrant une semaine ou un mois). Cette trace d'une activité anormale est généralement assez difficile à obtenir de manière aussi synthétique avec des outils orientés sécurité.



Suivi de l'activité UDP

Toutefois, un fichier RRDTool, par conception, ne contient pas toutes les données nécessaires à une analyse. Ce type d'alerte est donc limité : l'anomalie détectable doit être suffisamment importante par rapport à la fenêtre d'analyse que l'on considère (on doit donc avoir une activité anormale pendant plusieurs minutes pour l'historique sur une journée, mais pendant plusieurs heures pour l'historique sur une semaine par exemple). Enfin, il faut noter que la détection automatique des anomalies semble possible. Une fonction nouvelle de RRDTool qui est intégrée dans la version en développement de l'outil est entièrement dédiée à la détection de ces anomalies de comportement. Cette nouvelle fonction (nommée aberrant behaviour detection (http://cricket.sourceforge.net/aberrant/)) s'appuie sur le calcul de nouveaux ensembles de données correspondant à une modélisation statistique des valeurs attendues pour les données observées (en fonction de l'historique précédent) et l'identification d'une alerte en cas d'écart avec ce modèle. A notre sens, il devrait être très intéressant d'étudier la pertinence de cet indicateur de détection d'anomalies pour des systèmes informatiques soumis à des attaques.

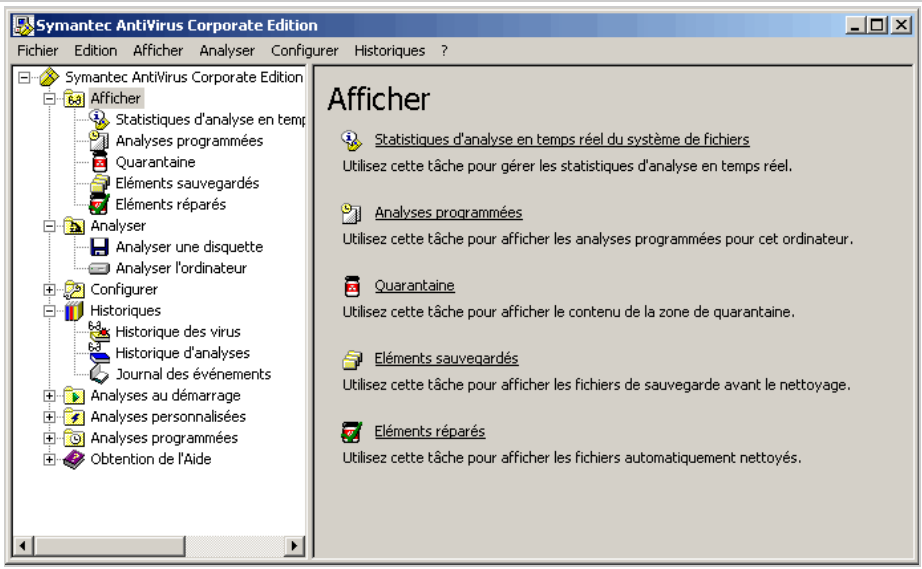
Sécurité informatique/Antivirus

Les systèmes antivirus font partie des systèmes les plus répandus dans le domaine de la sécurité informatique. De manière générale, il s'agit de systèmes de détection de codes malveillants basés sur la détection de signatures. Ces signatures de détection sont identifiées par les éditeurs d'antivirus qui les mettent à disposition de leurs clients, généralement via des systèmes automatiques de téléchargement. On trouve des logiciels antivirus utilisant ces signatures à plusieurs endroits dans le système informatique, notamment sur les postes de travail et sur les principales passerelles applicatives (messagerie, relais HTTP).

Poste de travail

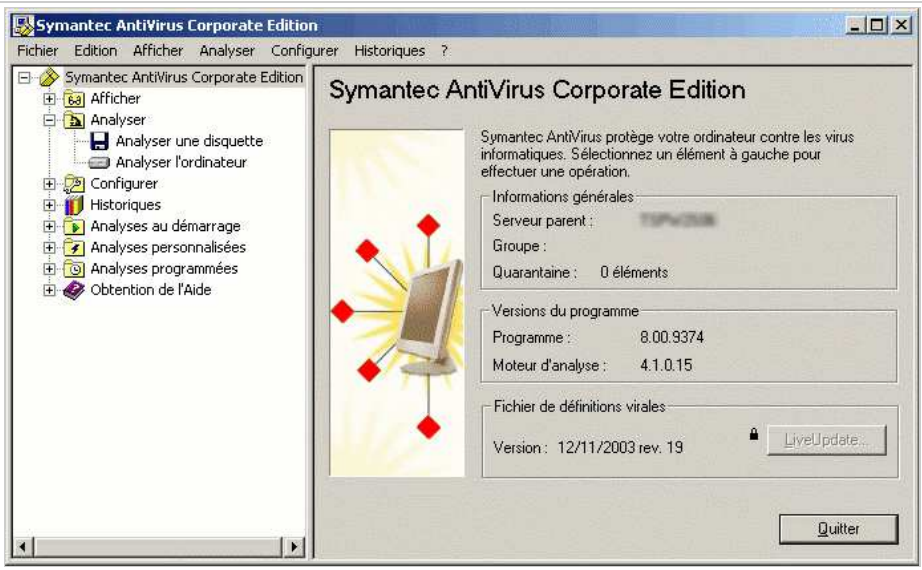
Initialement, les antivirus ont été déployés sur les postes de travail. Ils permettent ainsi de réaliser une analyse complète des éléments du système de fichiers afin de détecter d'éventuels virus présents dans les fichiers et de les éliminer soit en corrigeant, soit en détruisant le fichier concerné. Associés à des composants du système d'exploitation, ils permettent d'effectuer une analyse plus dynamique, souvent dite « en temps réel », des différents fichiers ouverts par les applications, de manière à détecter les virus au plus tôt, notamment dans le cas de leur diffusion au sein de documents. Les analyses complètes programmées (nocturnes) sont généralement utilisées sur les serveurs (pour lesquels une analyse dynamique poserait des problèmes de performance en raison du grand nombre de fichiers accédés) tandis qu'une analyse temps réel est généralement préférable sur les postes de travail.

Sur le poste de travail de l'utilisateur final, l'interface de l'antivirus permet d'afficher un certain nombre d'informations générales sur les détections éventuelles effectuées par l'antivirus (voir figure suivante) ainsi que de lancer manuellement des analyses complètes du système de fichiers à la demande. En général, l'utilisateur a peu de contrôle sur les paramètres de fonctionnement de l'antivirus qui restent réservés aux administrateurs et sont diffusés par le serveur de l'entreprise. Notamment, il est important qu'un utilisateur peu averti ne puisse pas désactiver facilement l'antivirus1. Dans le logiciel présenté en exemple, on note aussi la présence d'un système de quarantaine permettant d'isoler les fichiers contaminés par des virus pour des analyses ultérieures plus détaillées.



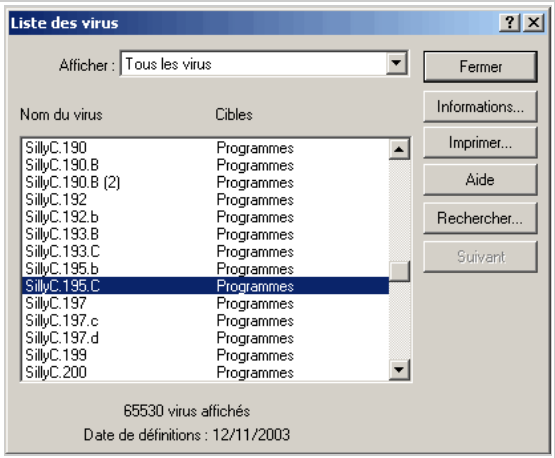
Fonctions de l'antivirus accessibles à l'utilisateur final

La figure suivante présente les principaux paramètres de l'antivirus du poste de travail. On y trouve d'abord identifié le serveur parent du poste de travail du point de vue de l'antivirus : c'est ce serveur qui distribue à ce poste les mises à jour de signatures permettant de détecter de nouveaux virus, ainsi que d'éventuelles évolutions du moteur d'analyse lui-même. Sont ensuite listés les différentes versions du programme, du moteur d'analyse, et surtout du fichier de signatures (nommé fichier de définitions virales dans l'exemple présenté). Afin de pouvoir détecter efficacement tous les virus connus et surtout les plus récents, ce fichier de signatures doit bien évidemment être aussi récent et aussi complet que possible.



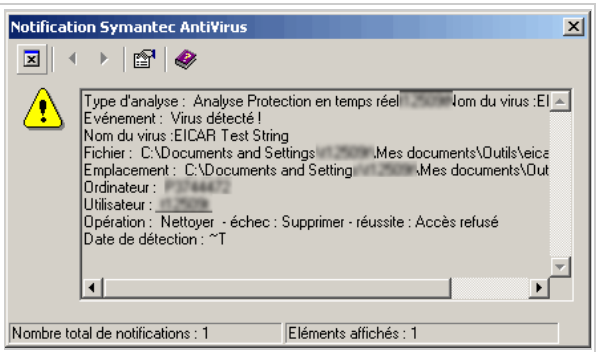
Interface principal de l'antivirus

La figure suivante donne un exemple de la liste des virus détectés correspondant au fichier de signatures de l'antivirus. On voit que, même en 2003, le nombre de signatures prises en compte était très important.



Liste des signatures de virus

Enfin, la figure suivante présente un exemple d'alerte émis en direction de l'utilisateur en cas de détection de virus. Il s'agit dans ce cas du virus de test présenté au §4.1.3. Parmi les informations affichées, on note l'action effectuée par l'antivirus : il s'agit en général soit d'une opération de correction du fichier (nettoyage), soit d'une suppression du fichier concerné, soit éventuellement d'une mise en quarantaine.



Exemple de message d'alerte

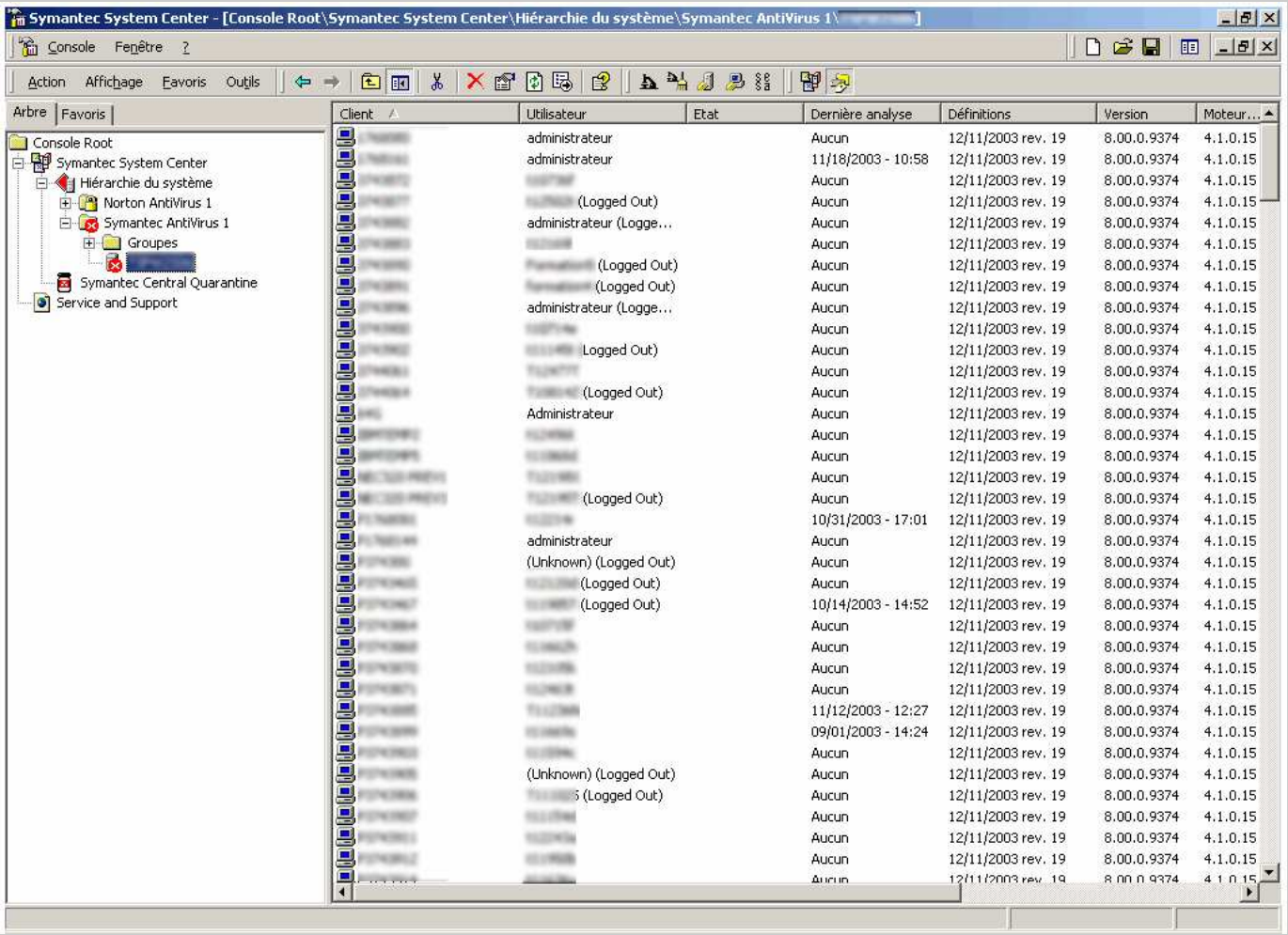
Il est possible de désactiver ces messages d'alerte (surtout s'ils sont aussi collectés par d'autres moyens). D'après notre expérience, le fait de les désactiver ou non est une question assez difficile à trancher.

En général, en cas d'infection virale, la plupart des utilisateurs même avertis, ont une vision très peu fiable de la réalité de l'infection de tel ou tel poste de travail. Ainsi, la présence d'une alerte de l'antivirus est en fait certainement le signe que le poste de travail n'a pas été infecté ; tandis que son voisin qui, lui, n'a vu aucune alerte, peut très bien avoir été corrompu (notamment s'il disposait d'un fichier de définitions virales un peu plus ancien). Dans un cas comme dans l'autre, caractériser précisément la présence ou l'absence d'un virus particulier demande une connaissance précise de ses caractéristiques techniques (les paramètres systèmes qu'il altère par exemple). L'alternative (notamment pour des informaticiens peu familier des problèmes de sécurité) est d'effectuer une analyse complète du système de fichiers après avoir vérifié la mise à jour des signatures utilisées par l'antivirus ; mais cette alternative n'est pas toujours aussi fiable qu'une étude manuelle, notamment du fait que certains virus récents commencent à essayer d'altérer le fonctionnement des antivirus courants.

Dans cette situation, le déclenchement des alertes conduit généralement les utilisateurs à compliquer la gestion d'une infection virale en occupant l'attention des administrateurs. Ceci tendrait à favoriser leur désactivation. Toutefois, en cas de détection d'un virus, il peut aussi sembler normal d'avertir l'utilisateur direct du poste de travail, qui est le premier concerné par les données mises en danger et qui peut ainsi comprendre le travail que réalise en permanence l'antivirus sur son poste.

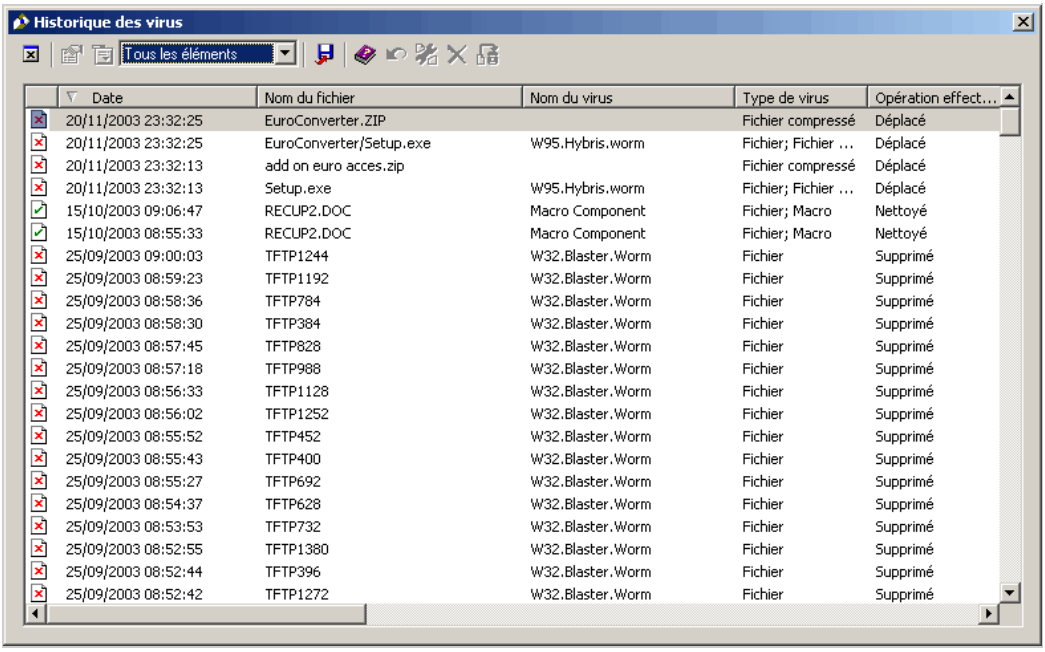
Console de gestion

Du point de vue de l'administrateur, les différents moyens de gestion de l'antivirus sont généralement centralisés autour d'une console de gestion (souvent associée au serveur de distribution du fichier des signatures). La figure suivante montre un exemple de la console de gestion du produit présenté précédemment (Symantec Antivirus). On y voit les différents serveurs utilisés pour la distribution du fichier des signatures, ainsi que les évènements correspondant aux connexions des postes de travail. On peut également y trouver identifiés individuellement les serveurs du système informatique pour lesquels on gère une configuration individualisée (fréquence et horaires des analyses programmées notamment).



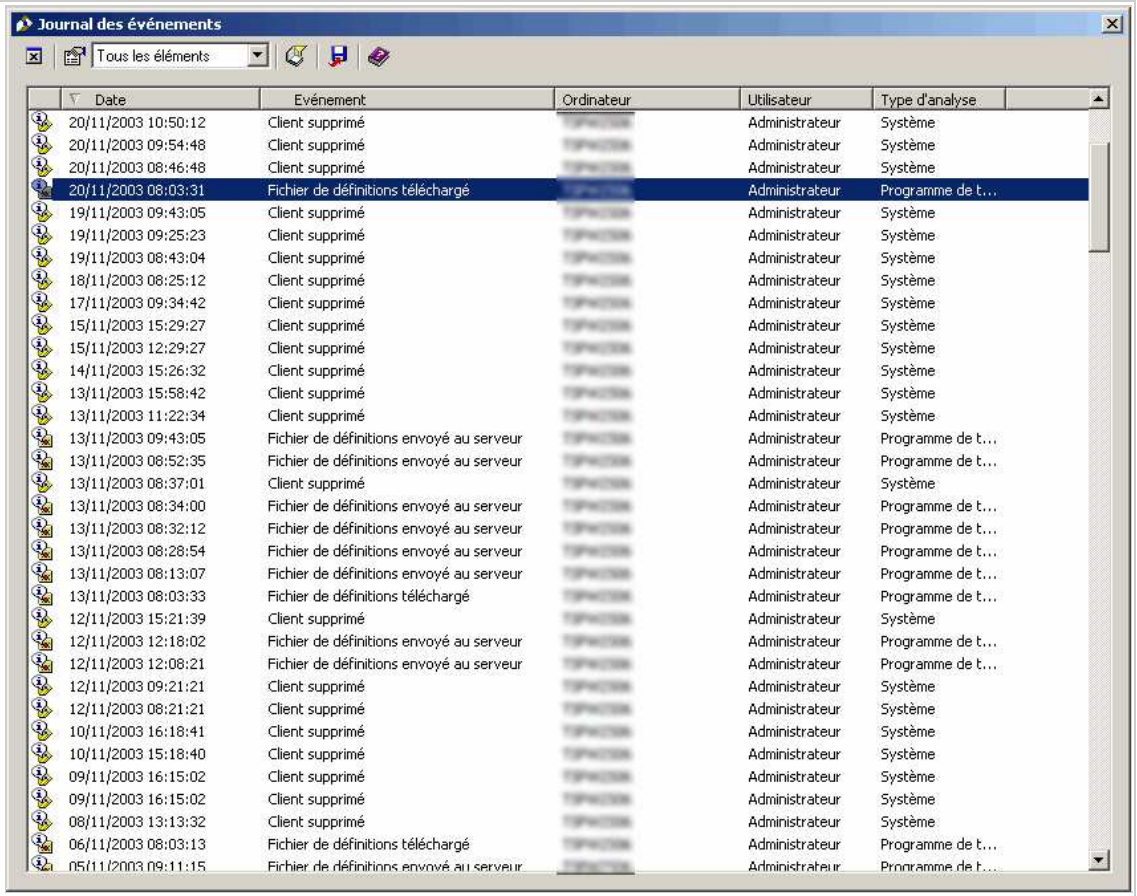
Console centralisée de gestion de l'antivirus

La console de gestion centralise également l'ensemble des alertes de détection de virus déclenchées dans l'entreprise. La figure suivante présente un exemple de cet historique coïncidant avec quelques tentatives de ré-infection du virus Blaster au moment du retour sur le réseau d'entreprise de postes de travail nomades infectés. (En temps normal, l'historique est beaucoup moins riche et, normalement, ne montre des détections de virus que de manière épisodique.)



Historique des virus détectés

Enfin, la dernière figure présente l'historique des évènements de gestion de l'antivirus, également proposé au niveau de la console de gestion. Cet historique identifie notamment tous les évènements de mise à jour des signatures pour les antivirus. Il faut garder à l'esprit le fait qu'un poste de travail peut prendre un certain retard par rapport à la dernière version disponible (par exemple si son utilisateur normal est absent pendant un certain temps). Par contre, il peut être utile d'identifier les postes de travail dont les mises à jour ne se font pas de manière normale, c'est à dire ceux qui sont absents de cet historique ou ceux dont les mises à jour échouent. (Suivant les produits, ces informations-là sont parfois assez difficiles à consolider; la dernière version du produit antivirus utilisé comme support à cette présentation fournissant notamment un outil de *reporting* dédié à cette consolidation.)



Evènements de gestion de l'antivirus

Tester un antivirus

Le test du bon fonctionnement d'un logiciel antivirus est un sujet plus délicat qu'il n'y paraît.

D'abord c'est un point important : un logiciel antivirus n'a d'intérêt que s'il est réellement en mesure de stopper des virus. Tout dysfonctionnement devrait être détecté.

Ensuite, il est absolument hors de question de tester le bon fonctionnement d'un antivirus en introduisant des virus réels dans un système informatique ! D'abord, il n'est pas forcément facile d'obtenir un virus réel, la préoccupation courante en sécurité informatique étant plutôt de les éradiquer que de les conserver. Par ailleurs, même si ce type de test était effectué sur un système isolé en salle blanche, totalement déconnecté de tout autre système, le risque resterait important de déclencher une propagation réelle (fut-ce par une erreur de manipulation). Par ailleurs, dans ce cas, l'intérêt du test lui-même serait assez limité car un système isolé ne permet pas de tester l'environnement d'exploitation réel (et notamment par exemple la remontée des alertes vers une console de gestion).

Pour pallier à ces difficultés de test du déploiement, la plupart des constructeurs d'antivirus ont implémenté une signature de détection d'un virus factice, nommé EICAR. La définition de référence de ce virus est accessible à l'URL suivante : http://www.eicar.org/anti_virus_test_file.htm. Ce virus est sans danger car il correspond tout simplement à un fichier texte contenant les 68 caractères suivants et ayant une longueur d'exactement 68 octets :

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

Ce fichier est par ailleurs un programme DOS exécutable valide dont le seul effet inoffensif est d'imprimer le message suivant : « EICAR-STANDARD-ANTIVIRUS-TEST-FILE! ».

Créer ou copier un tel fichier dans un espace de test sur le disque dur d'un système est donc un excellent moyen de tester le bon fonctionnement du logiciel antivirus. Par contre, il faut noter qu'en règle générale ce fichier sera immédiatement traité par le logiciel antivirus, c'est à dire soit effacé, soit verrouillé (et donc difficile à détruire une fois le test effectué).

Antivirus de flux : messagerie, flux HTTP (entrant)

Les virus utilisant d'autres moyens de propagation, notamment les messages électroniques et les flux HTTP (via les possibilités de téléchargement ou d'exécution de code offertes par les clients de messagerie et les navigateurs) de nouveaux besoins d'analyse antivirale sont apparus. Au niveau des passerelles de messagerie ou des relais HTTP, ces antivirus fonctionnent de manière assez similaire à ceux présents sur les postes de travail en s'appuyant sur des définitions de signatures. Le fonctionnement de l'antivirus est surtout différent en ce sens qu'il traite un flux de communication qu'il doit essayer de ne pas perturber, notamment du point de vue de ses performances.

En ce qui concerne les flux de messagerie, l'antivirus doit analyser les pièces jointes aux messages électroniques. Celles-ci pouvant être incluses dans des fichiers compressés, le travail d'analyse est conséquent et le dimensionnement des plate-formes matérielles réalisant ces analyses est assez délicat. En ce qui concerne les flux HTTP, la principale difficulté consiste à réaliser une analyse à la volée sans bloquer le flux de communication. Plusieurs stratégies sont envisageables et certaines peuvent modifier assez sensiblement le ressenti des utilisateurs finaux (notamment dans le cas où les fichiers sont stockés transitoirement par l'antivirus).

Enfin, tenter de télécharger depuis l'URL de référence un des fichiers disponibles contenant le virus factice EICAR est un bon moyen de tester la présence et le bon fonctionnement d'un antivirus HTTP s'il existe. Le test d'un antivirus de messagerie peut être effectué en envoyant le fichier dans un message électronique.

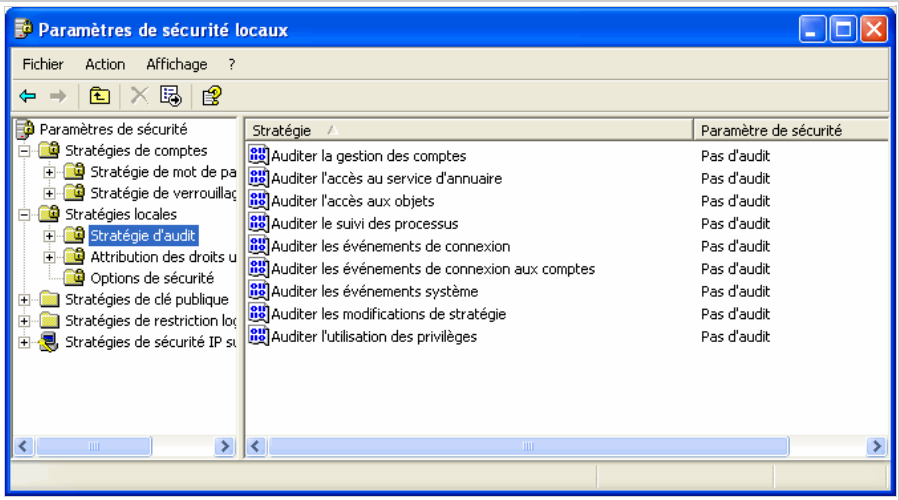
Sécurité informatique/Configuration des traces et mots de passe Windows

Configuration des traces Windows 2000 (et ultérieurs)

Parmi les différentes traces disponibles dans un système informatique, celles que l'on peut collecter sur les systèmes d'exploitation de la famille de Microsoft Windows 2000 ou XP sont particulièrement communes. On peut en effet généralement disposer de ces traces sur la majorité des postes de travail et une large part des serveurs d'un système informatique d'entreprise usuel. De plus, en y regardant de plus près, ces traces sont assez riches et relativement faciles à activer via les facilités d'administration offertes par l'annuaire centralisé Active Directory de Microsoft.

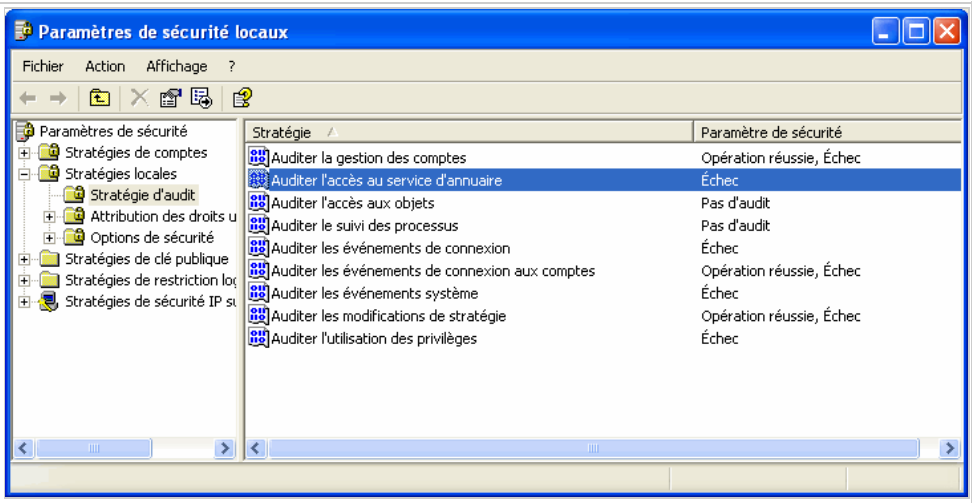
Pourtant, ces traces sont certainement sous-exploitées dans la majeure partie des configurations. En effet, leur disponibilité n'est pas toujours connu, leur analyse n'est pas évidente (elles sont très nombreuses) et leur centralisation n'est pas immédiatement réalisable via les seules fonctionnalités du système d'exploitation Microsoft. Pour rester dans un univers logiciel homogène, il est nécessaire d'acquérir d'autres solutions de l'éditeur (Microsoft Operations Manager ou MOM) pour la centralisation des traces. Toutefois, des solutions alternatives existent, en s'appuyant sur des logiciels plus spécifiques (et notamment le protocole et les serveurs syslog et le logiciel libre Ntsyslog (http://ntsyslog.sourceforge.net/)).

Avant de réaliser leur centralisation, il faut identifier les principaux paramètres de configuration des traces sur le système d'exploitation. Ceux-ci figurent globalement dans la « stratégie d'audit » du système d'exploitation, parmi les « stratégies locales » de sécurité. La figure suivante présente les différents paramètres disponibles ainsi que leur valeur par défaut (dans le contexte graphique de Windows XP). On note que par défaut cet audit est désactivé.



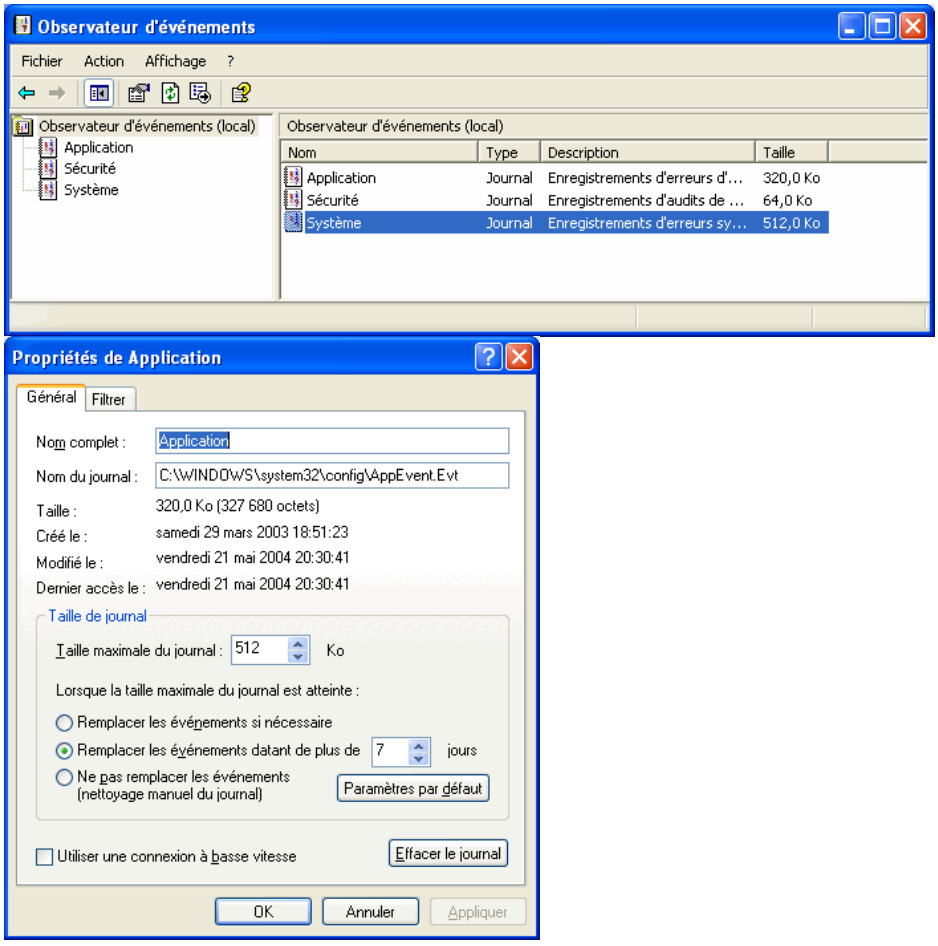
Configuration par défaut de l'audit Windows (XP)

Dans la figure suivante, nous présentons un exemple de configuration recommandée pour l'audit Windows de notre point de vue. Il s'agit là d'une configuration assez exhaustive, susceptible d'être restreinte en cas de problèmes de performance, mais qui donne satisfaction dans la plupart des cas. La configuration optimale dépend du système informatique concerné.



Configuration souhaitable de l'audit Windows (XP)

D'autres paramètres à prendre en compte pour la configuration de l'audit Windows concernent les fichiers journaux utilisés pour le stockage des traces. Ils sont présentés ci-aprés. Il faut notamment indiquer la taille maximale des fichiers journaux ainsi que la stratégie de remplacement à utiliser quand les journaux sont pleins. Les deux paramètres sont en fait étroitement liés.

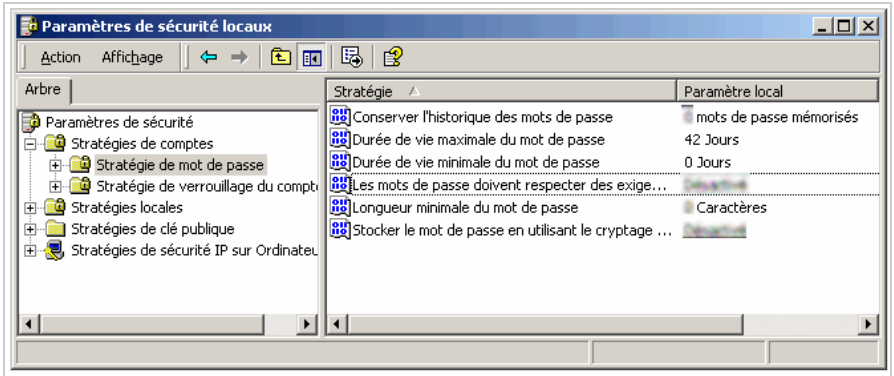


Si une taille importante (de l'ordre de la dizaine de MiB) est utilisable ; le fichier journal pourra certainement contenir l'ensemble des traces normalement générées pendant la période de rétention désirée. Dans ce cas, une stratégie de remplacement basée sur l'âge des traces est à notre sens préférable. En effet, elle évite de laisser à un attaquant l'opportunité de masquer ses traces en saturant le fichier journal par un événement anodin répété suffisamment pour entraîner une rotation rapide. Ceci correspond globalement à une politique consistant à conserver les traces sur la machine sur laquelle elles sont collectées.

Si on préfère réserver une taille limitée pour le fichier journal sécurité et si celui-ci n'est utilisé que comme tampon intermédiaire avant déport des traces vers un serveur de centralisation, il est alors préférable d'utiliser une taille très réduite pour le journal (quelques dizaines de KiB) et d'adopter une stratégie de remplacement au besoin pour éviter qu'un pic d'activité (normal) de la machine ne conduise à une perte des traces. Enfin, il faut penser que, dans la plupart des cas, l'activation des traces ne doit pas se faire à l'aveuglette. La génération, le stockage et la destruction de ces journaux doit être maîtrisée. D'abord parce que, par exemple, un serveur syslog mal configuré est un des meilleurs moyens de remplir rapidement un disque dur (intentionnellement ou non). Ensuite parce que, en France, la protection des données personnelles inclut une obligation de protéger ces données et l'utilisation qui en est faite de manière effective ainsi qu'un droit à l'oubli dont l'esprit est de prononcer la destruction systématique de données personnelles après leur péremption (surtout si elles ne sont pas utilisées). Une bonne maîtrise du cycle de vie des journaux est donc nécessaire avant de s'engager dans leur activation.

Par ailleurs, la collecte et même la centralisation des journaux ne sont qu'une première étape, qui fournit avant tout un garde-fou en cas d'enquête approfondie. C'est l'exploitation de ces traces qui permet de progresser dans la gestion de la sécurité.

Gestion des mots de passe sous Windows



Paramètres de gestion des mots de passe sous MS/Windows

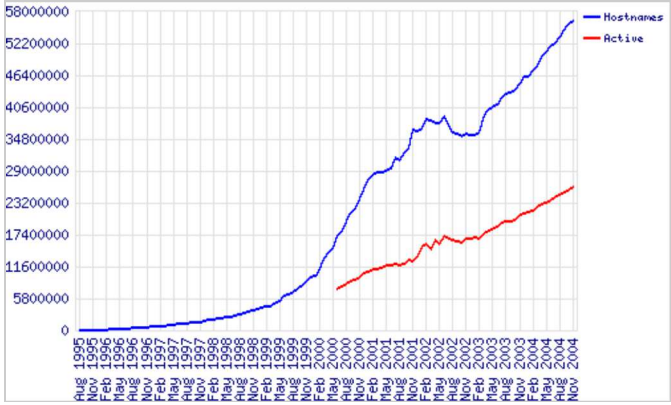
La figure précédente présente les différents paramètres disponibles sous MS/Windows pour imposer des contraintes supplémentaires aux utilisateurs en ce qui concerne le choix et la durée de vie de leurs mots de passe. Certains de ces paramètres peuvent aider à la mise en place effective d'une politique de sécurité. Toutefois, compte tenu de la difficulté d'amener les utilisateurs à utiliser des mots de passe solides ces paramètres doivent être utilisés avec précaution.

Sécurité informatique/Serveur HTTP

L'avènement du protocole HTTP et du Web, étroitement associée à l'explosion de l'utilisation d'Internet, a été une nouvelle étape du développement de l'utilisation de l'informatique et des réseaux par un nombre grandissant d'utilisateurs (voir ci-contre et les données actualisées (http://news.netcraft.com/archives/web_server_survey.html)). Au cœur de cette fonction se trouve bien évidemment le serveur HTTP lui-même, et notamment une de ses implémentations les plus répandues : le logiciel libre Apache.

De plus en plus d'applications s'appuient aussi sur le protocole HTTP et les serveurs associés pour gérer les interactions avec leurs utilisateurs. Ces applications Web vont parfois jusqu'à réimplémenter au-dessus du protocole HTTP des infrastructures de fonctionnement complètes (par exemple de type RPC avec SOAP) qui mettent le serveur HTTP au cœur de leur exécution.

La sécurité des serveurs Web (c'est à dire des serveurs HTTP et HTTPS) est donc devenu un enjeu majeur. Dans cette section, nous essayons d'en aborder certains aspects concrets, sans nous soucier de la sécurité sur le Web au sens large, mais pour mettre l'accent sur les fonctions du logiciel présent au centre du système et dont les paramètres et les fonctions de sécurité nous semblent malgré tout assez méconnues et pas toujours bien maîtrisées.



Evolution du nombre de sites Web 1995-2004 (source Netcraft (http://news.netcraft.com/))

D'abord, stricto senso, notamment du point de vue réseau il faut faire la distinction entre le serveur HTTP (TCP/80) et le serveur HTTPS (TCP/443). Il s'agit en général du même programme, offrant souvent un accès aux mêmes données, mais en utilisant soit une connexion TCP conventionnelle, soit une connexion incluse dans SSL/TLS. Ce dernier protocole offre des fonctions de sécurité avancées pour les communications point à point entre client et serveur, pouvant aller jusqu'à une authentification mutuelle utilisant des techniques de cryptographie asymétrique avec des longueurs de clefs supérieures à 1024 bits et une protection de l'intégrité et de la confidentialité de la communication avec des algorithmes symétriques variés et des longueurs de clefs de session tout à fait acceptable (56, 64 ou 128 bits). Toutes les possibilités de SSL/TLS ne sont généralement pas exploitées dans les configurations courantes, notamment l'authentification du client.

Pourtant, pour la sélection des utilisateurs, on peut envisager deux grandes approches, avec des niveaux de protection bien différents :

- La sélection peut-être basée sur des plages d'adresses IP, à l'instar des contrôles effectués par un firewall. Le serveur HTTP(S) décide d'autoriser l'accès à une URL particulière en fonction de l'adresse du client.
- Via HTTPS seulement, la sélection peut également s'appuyer sur un certificat X.509 présenté par le navigateur client (relié au certificat du serveur) et l'identité garantie par ce certificat.

En ce qui concerne les URL (ou plus généralement les ensembles d'URL) accessibles via un serveur, la sélection des destinations et de la manière dont on peut y accéder peut s'effectuer également de deux grandes manières :

- soit en s'appuyant sur le chemin d'accès (URL ou nom dans le système de fichier) ;
- soit en analysant le type d'extension de l'URL accédée (par exemple .html pour des fichiers simples, .php, .jsp, .asp pour des pages dynamiques, etc.).

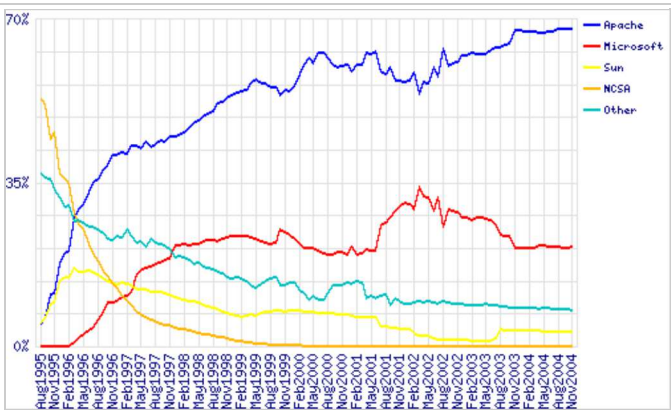
S'agissant des pages dites « dynamiques », celles-ci sont obtenues non pas par un simple accès au système de fichiers, mais par l'exécution d'un programme secondaire générant la page. Ce programme est parfois exécuté dans le contexte même du processus du serveur, lequel doit alors pré-charger ou se lier dynamique à des exécutables externes adaptés (des modules). La maîtrise des extensions dynamiques, de leur présence ou de leur absence et de leurs configurations spécifiques, est un point important de la sécurité du serveur HTTP. Beaucoup de serveurs incluent certainement des modules dynamiques dont ils n'ont pas besoin, et offrent donc des fonctions d'accès inutilisées et presque inconnues de leurs administrateurs.

On le voit, les serveurs HTTP sont des composants logiciels plus complexes qu'il n'y paraît ou en tout cas dont l'environnement d'exécution est complexe. La sécurité du serveur peut passer par un meilleur contrôle du processus serveur lui-même au niveau du système d'exploitation visant à limiter les débordements possibles via ce processus. On peut alors envisager deux grandes voies pour mieux contrôler ce processus critique :

- l'utilisation de techniques de confinement (notamment sous Unix via chroot - un effort notamment réalisé dans OpenBSD (http://www.openbsd.org/faq/faq10.html#httpdchroot)) ;
- la mise en place réfléchie de restrictions d'accès en lien avec le système de fichiers (arborescences www bien limitées et au contenu validé).

Apache 1.3

Le serveur HTTP librement disponible du projet Apache (http://httpd.apache.org) est une des principales implémentations de serveur HTTP utilisée dans la pratique (voir graphique).



Répartitions des principaux serveurs (source Netcraft (http://news.netcraft.com/))

En parcourant un fichier de configuration standard d'Apache, notamment ceux de la distribution Debian GNU/Linux dont le fonctionnement est très réfléchi, on distingue un certain nombre de directives qu'il semble souhaitable de contrôler dans la perspective de la sécurité du service.

Les fichiers de configuration d'Apache sont nommés par défaut httpd.conf, access.conf et srm.conf (on rencontre également le nom apache.conf à la place du premier, et la présence d'un fichier nommé modules.conf est aussi assez fréquente pour isoler les directives de chargement des modules). Les différents fichiers sont normalement associés à des directives différentes. Toutefois, ces fichiers ne sont pas les seuls pouvant être impliqués dans la configuration complète : le langage de configuration est modulaire via l'utilisation de directives Include et cette modularité est souvent mise à profit pour isoler les directives de configuration des modules dans des fichiers séparés.

Dans le fichier principal par défaut (httpd.conf en général) dont nous présentons

certains extraits ici, les directives mentionnées en commentaire (débutant par un #) indiquent les valeurs par défaut utilisées par Apache en l'absence de la directive (supprimer simplement le # n'a donc normalement aucun effet sur le serveur).

Certaines directives concernent la configuration réseau fondamentale, c'est à dire l'adresse IP et le port TCP de la socket ouverte par le serveur lors de son démarrage. Les directives BindAddress et Port permettent d'indiquer l'adresse IP et le numéro de port à utiliser. Mais la première est obsolète et devrait être éliminée dans la version 2.0 d'Apache. Listen est une forme plus moderne permettant de spécifier en une seule fois les paramètres réseau du port de communication, en séparant l'adresse IP et le numéro de port par : . En l'absence d'adresse IP spécifiée (ou si BindAddress * est utilisé) le serveur Apache écoutera sur toutes les adresses IP disponibles, y compris des adresses virtuelles ou des adresses correspondant à plusieurs cartes réseau. C'est notamment dans ces deux cas qu'une directive explicite est souhaitable pour éviter les confusions. Des directives Listen explicites sont également indispensables (éventuellement sous la forme Port) pour démarrer un serveur Apache fournissant un ou plusieurs services sur un ou des ports différents du port HTTP standard (80).

```
#Listen 3000
#Listen 12.34.56.78:80
#BindAddress *
Port 80
```

Le fonctionnement du serveur Apache est largement basé sur les possibilités offertes par une architecture logicielle modulaire. La plupart des fonctions additionnelles du serveur sont fournies par des modules. Même certaines qui paraissent naturelles du point de vue de l'utilisateur sont en fait des extensions par rapport aux services élémentaires fournis par le serveur HTTP (comme le contrôle d'accès, SSL/TLS, les programmes CGI externes, ou certaines directives de configuration comme les alias de chemins d'accès). Ces modules sont chargés par Apache quand il rencontre une directive LoadModule dans le fichier de configuration. Ces directives arrivent assez tôt dans la configuration car elles peuvent étendre le langage de configuration en offrant de nouvelles directives. L'identification précise de tous les modules chargés par Apache est cruciale du point de vue de la sécurité, car c'est elle qui permet de savoir quel est exactement le périmètre fonctionnel offert par un serveur particulier. Par exemple, avec les directives suivantes, le serveur Apache, outre qu'il est un serveur HTTP, offre également : des fonctions permettant d'exécuter des programmes externes (CGI) générant des pages Web (mod_cgi), un interpréteur intégré du langage PHP (libphp4), des directives de contrôle d'accès (mod_access) et une directive de configuration permettant de définir des Alias (mod_alias) de chemin d'accès pour les URL (pour repositionner différentes parties du système de fichiers dans l'arbre des documents accessibles ou permettre la redirection des URL).

```
LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so
# LoadModule asis_module /usr/lib/apache/1.3/mod_asis.so
LoadModule alias_module /usr/lib/apache/1.3/mod_alias.so
LoadModule access_module /usr/lib/apache/1.3/mod_access.so
LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
```

Les directives User et Group permettent de définir l'identifiant utilisateur et l'identifiant du groupe (UID et GID sous Unix) que doivent posséder les processus serveurs (démons httpd). Le processus initial d'Apache s'exécute en effet généralement avec les droits du super-utilisateur (root sous Unix) afin d'initialiser les ports de communication nécessaires situés sur des ports TCP privilégiés, mais les processus serveurs répondant aux requêtes des navigateurs peuvent eux s'exécuter avec des identifiants non-privilégiés. Dans l'exemple suivant, ceux-ci vont s'exécuter avec des UID et GID dédiées nommées www-data.

```
User www-data
Group www-data
```

Compte tenu de la modularité du logiciel et de la possibilité pour les modules de fournir de nouvelles directives de configuration, des éléments de configuration conditionnelle sont disponibles, notamment via la directive <IfModule>. Sa syntaxe est particulière car elle nécessite, pour délimiter des blocs, des marqueurs de début et de fin comme dans l'exemple ci-après. Ces directives permettent d'écrire des portions de configuration valides même si le module concerné n'est pas chargé au démarrage (via une directive LoadModule). Elles sont surtout intéressantes pour préparer certains paramétrages dans une configuration type dans le cas où la mise en place des modules concernés est décidée ultérieurement suivant les besoins des utilisateurs du serveur Web. L'activation ou la désactivation d'un ou plusieurs modules peut alors être envisagée avec une relative sérénité durant l'exploitation. Dans l'exemple que nous prenons, le module mod_status est un module permettant d'afficher des informations internes concernant le serveur Apache lui-même. A priori, cette information n'est pas nécessaire aux utilisateurs normaux du site (c'est une information surtout intéressante pour ses administrateurs) et il serait également nécessaire de limiter l'accès à ces informations.

```
<IfModule mod_status.c>
  ExtendedStatus On
</IfModule>
```

Outre les directives de configuration conditionnelle, la directive Include permet de choisir une configuration modulaire, séparée en plusieurs fichiers. Les exemples ci-dessous illustrent le principe de la distribution Debian dans laquelle des portions de configuration d'Apache concernant exclusivement certaines applications Web sont en fait localisées aux côtés des fichiers de configuration de l'application elle-même. Chacun de ces fichiers contient des paramétrages du serveur HTTP (notamment des règles de contrôle d'accès) appliqués uniquement à la portion de l'arborescence des documents correspondant à l'application concernée.

```
Include /etc/phpmyadmin/apache.conf
Include /etc/phpgroupware/apache.conf
```

Afin d'adapter les paramètres aux besoins des différentes parties de l'arborescence des documents accessibles, Apache offre un certain nombre de directives que nous qualifions de « contexte ».

```
<Directory> et <DirectoryMatch>
<Files> et <FilesMatch>
<Location> et <LocationMatch>
<VirtualHost>
```

Ces directives qui servent comme <IfModule> à identifier des blocs de lignes permettent d'appliquer des paramètres uniquement dans certains contextes : soit pour l'accès à certaines répertoires du système de fichier (<Directory> et <DirectoryMatch>), soit pour l'accès à certains fichiers (<Files> et <FilesMatch>), soit pour des accès effectués via certaines URL (<Location> et <LocationMatch>). Par ailleurs, Apache offre la possibilité de définir des serveurs virtuels via la directive <VirtualHost> et de leur appliquer des règles de configuration spécifiques. Ceux-ci se comportent comme des serveurs indépendants associés à des noms de domaine différents tandis qu'ils correspondent en fait seulement à des parties différentes de l'arborescence de documents d'un même serveur Apache (les différents noms de domaine doivent toutefois pointer vers une même machine - généralement via un alias DNS).

Le contrôle des chemins d'accès (via les répertoires) permet ainsi de proposer des configurations de ce type :

```
<Directory />
    Options SymLinksIfOwnerMatch
    AllowOverride None
</Directory>
…
<Directory /var/www/>
 Options Indexes Includes FollowSymLinks MultiViews
 AllowOverride None
 Order allow,deny
    Allow from all
</Directory>
…
<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>
```

Dans cet exemple, le répertoire par défaut (racine /) se voit d'abord affecté une configuration très restrictive. Le répertoire /var/www est rendu accessible à tous, un certain nombre d'options sont également activées. Les répertoires public_html des utilisateurs (/home/*/public_html) sont aussi rendus accessibles à tous mais les utilisateurs peuvent modifier les règles de contrôle d'accès s'ils le souhaitent car la directive AllowOverride le leur permet (via des fichiers .htaccess). Par contre, les requêtes autorisées vers ces zones gérées directement par les utilisateurs sont limitées à des accès simples de consultation HTTP.

Le contrôle des noms de fichiers permet de traiter certains cas particuliers d'autorisation. Ainsi par exemple, pour interdire la consultation via le serveur HTTP des fichiers .htaccess permettant (éventuellement) de modifier les autorisations d'accès localement dans une zone du système de fichiers, la directive suivante peut être utilisée.

```
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
```

En combinant l'utilisation des alias (directive Alias) et le contrôle d'accès basé sur l'URL utilisée par le navigateur (directive Location), il est possible d'autoriser à certains clients particuliers l'accès à des zones du système de fichiers qui ne sont pas directement visible dans l'arborescence des documents normale (/var/www dans nos exemples). Par exemple, voici comment rendre accessible la documentation installée sur le système de fichiers dans /usr/share/doc via une URL du type http://mon.serveur/doc/. S'agissant de fichiers normaux (par exemple des fichiers textes), la gestion automatique d'index listant le contenu des répertoires est évidemment utile dans ce cas (directive Options Indexes).

```
Alias /doc/ /usr/share/doc/
<Location /doc>
  order deny,allow
  deny from all
  allow from 127.0.0.0/255.0.0.0
  allow from AA.BB.CC.0/255.255.XX.0
  Options Indexes FollowSymLinks MultiViews
</Location>
```

Voici comment permettre l'exécution de pages dynamiques (générées via des programmes en langage Perl) dans une arborescence spécifique de la zone du système de fichiers consacrée au Web. C'est la directive Options +ExecCGI qui permet l'exécution de programmes externes du type CGI, les programmes eux-mêmes étant exécutés par le handler spécifique au langage Perl (fournit par le module mod_perl s'il est chargé).

```
# If the perl module is installed, this will be enabled.
<IfModule mod_perl.c>
  Alias /perl/ /var/www/perl/
  <Location /perl>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
  </Location>
</IfModule>
```

Voici maintenant des exemples d'utilisation des directives de configuration que nous avons vues dans l'objectif de réaliser un contrôle d'accès à des pages dynamiques installées dans une arborescence spécifique (a priori non-publique) via des chemins d'accès. Il s'agit là de permettre d'utiliser l'interface de visualisation de Prelude-IDS écrite en Perl nommée Piwi.

```
# For Prelude PIWI
Alias /piwi /home/xxxx/prelude/piwi
ScriptAlias /piwi /home/xxxx/prelude/piwi
<DirectoryMatch /home/xxxx/prelude/piwi/>
 order allow,deny
 allow from all
 Options +ExecCGI
 AddHandler cgi-script .pl
 DirectoryIndex index.pl
</DirectoryMatch>
```

Ce type de configuration d'une application Web est celui utilisé par la distribution Debian qui permet aussi d'isoler dans des fichiers de configuration secondaires les paramètres associés à une application Web par ailleurs installée dans une zone privée du système de fichiers. Cette approche nous semble intéressante du point de

vue de la sécurité, car elle permet vraiment de s'appuyer sur les directives de contrôle d'accès disponibles via le module mod_access pour gérer les autorisations d'accès à chaque application avec une bonne granularité et sans craindre trop d'interactions entre les différentes applications. Voici donc un exemple de fichier de configuration secondaire pour une application Web (PHPGroupware) sur ce système : /etc/phpgroupware/apache.conf

```
Alias /phpgroupware /usr/share/phpgroupware
<Directory /usr/share/phpgroupware/>
	Options +FollowSymLinks
	AllowOverride None
	order allow,deny
	allow from all
	DirectoryIndex index.html index.php
	<IfModule mod_php3.c>
		php3_magic_quotes_gpc On
		php3_track_vars On
		php3_include_path .:/etc/phpgroupware
	</IfModule>
	<IfModule mod_php4.c>
		php_flag magic_quotes_gpc On
		php_flag track_vars On
		php_flag session.save_path /var/tmp/phpgroupware
		php_value include_path .:/etc/phpgroupware
	</IfModule>
</Directory>
```

On notera également sur cet exemple la gestion simultanée de paramètres de configuration pour deux versions du module mod_php.

Apache et SSL (Apache-SSL ou Apache+mod_ssl)

On notera qu'Apache n'inclut pas dans la distribution de base du logiciel l'implémentation du protocole HTTPS, utilisant SSL/TLS6. L'installation d'un serveur HTTPS nécessite l'utilisation d'une des deux principales variantes d'Apache incluant le support de HTTPS :

- Apache+mod_ssl : implémentation utilisant un module d'Apache nommé mod_ssl (http://www.modssl.org) pour fournir le protocole HTTPS en s'appuyant sur l'implémentation OpenSSL de SSL/TLS.
- Apache-SSL (http://www.apache-ssl.org) : une implémentation d'origine plus ancienne focalisée sur la stabilité du logiciel mais également basée sur Apache et sur SSLeay/OpenSSL.

Le premier correspond à une division du développement du premier, maintenant suffisamment ancienne pour que les deux implémentations soient désormais réellement différentes.

Sécurité informatique/Filtrage d'URL

Parmi les relais utilisables pour assurer des fonctions de sécurité, les relais HTTP font partie des plus répandus étant donné la forte utilisation du protocole HTTP. Un des intérêts de disposer d'un relais HTTP est notamment de pouvoir y associer un logiciel dit de filtrage d'URL. L'objectif de ce filtre est de contrôler les accès HTTP sortants et si besoin d'interdire des URL correspondant à des sites indésirables. En effet, dans un contexte professionnel par exemple, le contrôle de l'accès à certains sites doit pouvoir être mis en œuvre pour éviter les débordements. Mais dans un contexte moins étroit, on peut aussi vouloir interdire l'accès à certaines catégories de sites à des enfants chez soi ou dans une école par exemple. Enfin, du point de vue de la sécurité, des contraintes légales (sites étrangers violant les lois françaises) ou des besoins techniques (sites hébergeant des codes malveillants) conduisent à se doter d'un moyen de contrôle permettant de limiter les accès.

Une des principales difficultés dans la réalisation du filtrage d'URL est de classifier de manière fiable les différents sites Web présents sur Internet et leurs URL. Cet effort de classification à d'abord été orienté en direction des sites associés à des catégories de caractère très marqué : pornographique, violent, piratage/malveillance, publicité, drogue, etc.

Les éditeurs de logiciels de filtrage commerciaux ont poursuivi cet effort de manière à atteindre une classification plus fine de l'ensemble des sites Web permettant, par certains côtés, de mettre en place des autorisations d'accès thématiques. On trouve ainsi des catégories du type « Éducation », « Gouvernement », « Jeux », « Recherche d'emploi », « Sport », « Voyages », etc. Ces catégories sont parfois divisées en sous-catégories, par exemple pour faire la différence entre des sites consacrés au piratage et ceux diffusant de l'information sur la sécurité informatique. Un tel effort de classification n'est pour l'instant disponible que parmi des solutions commerciales, dont la plus répandue semble être le logiciel Websense mais parmi lesquels on trouve également CyberPatrol ou Sentian.

Nous nous intéresserons tout d'abord plus en détail à la mise en œuvre d'un tel filtrage à l'aide de logiciels libres et notamment à l'aide du cache Squid et du logiciel de filtrage associé SquidGuard.

Squid

Squid (http://www.squid-cache.org/) est un des premiers et des principaux relais HTTP mis en place dans l'infrastructure Web d'Internet. L'objectif initial de ce relais est de fournir des fonctions de cache permettant de réduire les communications vers Internet en stockant les informations les plus demandées dans un cache proche des utilisateurs. Cette fonction d'apparence simple a été implémentée de manière assez avancée dans Squid dans l'objectif de pouvoir déployer des hiérarchies de caches offrant des performances accrues. Squid supporte complètement le protocole de communication inter-caches ICP [RFC 2186] permettant à des caches voisins ou proches parents dans la hiérarchie de se tenir informé du contenu des autres caches proches d'eux de manière à optimiser l'utilisation de l'ensemble des espaces de stockage affectés à cette tâche. Squid offre également une large palette de fonctions comme l'authentification des clients, la redirection, l'intégration avec des outils de filtrage, la surveillance via SNMP, un cache DNS, un mode d'accélération de site Web, etc. L'intérêt pratique des caches HTTP du point de vue des performances a largement décru avec l'apparition de plus en plus courante de contenus dynamiques sur les sites Web d'Internet, mais ils constituent néanmoins un outil indispensable dans la boite à outils des administrateurs Web, réseau ou sécurité, notamment en raison du filtrage qu'ils rendent possible.

Squid permet tout d'abord de limiter les clients autorisés à l'accéder en fonction de leur adresse IP. Ce contrôle est réalisé à l'aide de directives de configuration suivant les caractéristiques indiquées ci-dessous :

- Les directives ont deux composantes :
  - des éléments (ACL elements) ;
  - et des règles (access lists rules).
- Il est possible de combiner ces éléments avec des règles logiques :

```
acl_type {allow|deny} acl AND acl AND ...
OR acl_type {allow|deny} acl AND acl AND ...
OR ...
```

- Enfin, les exemples suivants montrent comment sont rédigées les autorisations :

```
acl all src 0/0
  http_access deny all
acl myclients src 1.2.3.0/24
  http_access allow myclients
```

Les informations suivantes extraites de la documentation d'origine de Squid détaillent l'ensemble des éléments utilisables pour la configuration des accès :

```
 Squid knows about the following types of ACL elements3 :
src: source (client) IP addresses
dst: destination (server) IP addresses
myip: the local IP address of a client's connection
srcdomain: source (client) domain name
dstdomain: destination (server) domain name
srcdom_regex: source (client) regular expression pattern matching
dstdom_regex: destination (server) regular expression pattern matching
time: time of day, and day of week
url_regex: URL regular expression pattern matching
urlpath_regex: URL-path regular expression pattern matching, leaves out the protocol and hostname
port: destination (server) port number
myport: local port number that client connected to
proto: transfer protocol (http, ftp, etc)
method: HTTP request method (get, post, etc)
browser: regular expression pattern matching on the request's user-agent header
ident: string matching on the user's name
ident_regex: regular expression pattern matching on the user's name
src_as: source (client) Autonomous System number
dst_as: destination (server) Autonomous System number
proxy_auth: user authentication via external processes
proxy_auth_regex: user authentication via external processes
snmp_community: SNMP community string matching
maxconn: a limit on the maximum number of connections from a single client IP address
req_mime_type: regular expression pattern matching on the request content-type header
arp: Ethernet (MAC) address matching
rep_mime_type: regular expression pattern matching on the reply (downloaded content) content-type header. This is only usable in th
```

```
external: lookup via external acl helper defined by external_acl_type »
```

Une fois définis, ces éléments peuvent être utilisés pour accorder ou non des accès en utilisant les types d'ACL suivants :

```
 There are a number of different access lists :
http_access: Allows HTTP clients (browsers) to access the HTTP port. This is the primary access control list.
http_reply_access: Allows HTTP clients (browsers) to receive the reply to their request. This further restricts permissions given b
icp_access: Allows neighbor caches to query your cache with ICP.
miss_access: Allows certain clients to forward cache misses through your cache. This further restricts permissions given by http_ac
no_cache: Defines responses that should not be cached.
redirector_access: Controls which requests are sent through the redirector pool.
ident_lookup_access: Controls which requests need an Ident lookup.
always_direct: Controls which requests should always be forwarded directly to origin servers.
never_direct: Controls which requests should never be forwarded directly to origin servers.
snmp_access: Controls SNMP client access to the cache.
broken_posts: Defines requests for which squid appends an extra CRLF after POST message bodies as required by some broken origin se
cache_peer_access: Controls which requests can be forwarded to a given neighbor (peer). »
```

SquidGuard

En utilisant les mécanismes de redirection disponibles au niveau de Squid, il est possible de développer un redirecteur jouant le rôle d'un outil de validation et de contrôle d'accès des URL accédées. C'est exactement ce que réalise le logiciel SquidGuard (http://www.squidguard.org/), en utilisant des techniques particulières pour gérer des listes de grande taille (plus de 100 000 entrées pour la catégorie *porn* par exemple) le plus efficacement possible.

La configuration de SquidGuard passe également par la définition de listes de contrôle d'accès indiquant pour certaines population d'utilisateurs les règles de filtrage à appliquer. On notera que ces règles peuvent prendre en compte des plages horaires pour modifier les autorisations d'accès en fonction du moment de la journée.

Enfin, le site de SquidGuard propose un ensemble de listes noires vérifiées périodiquement ; mais dont la mise à jour est essentiellement effectuée manuellement.

Voici un exemple assez facile à comprendre de configuration utilisable avec SquidGuard :

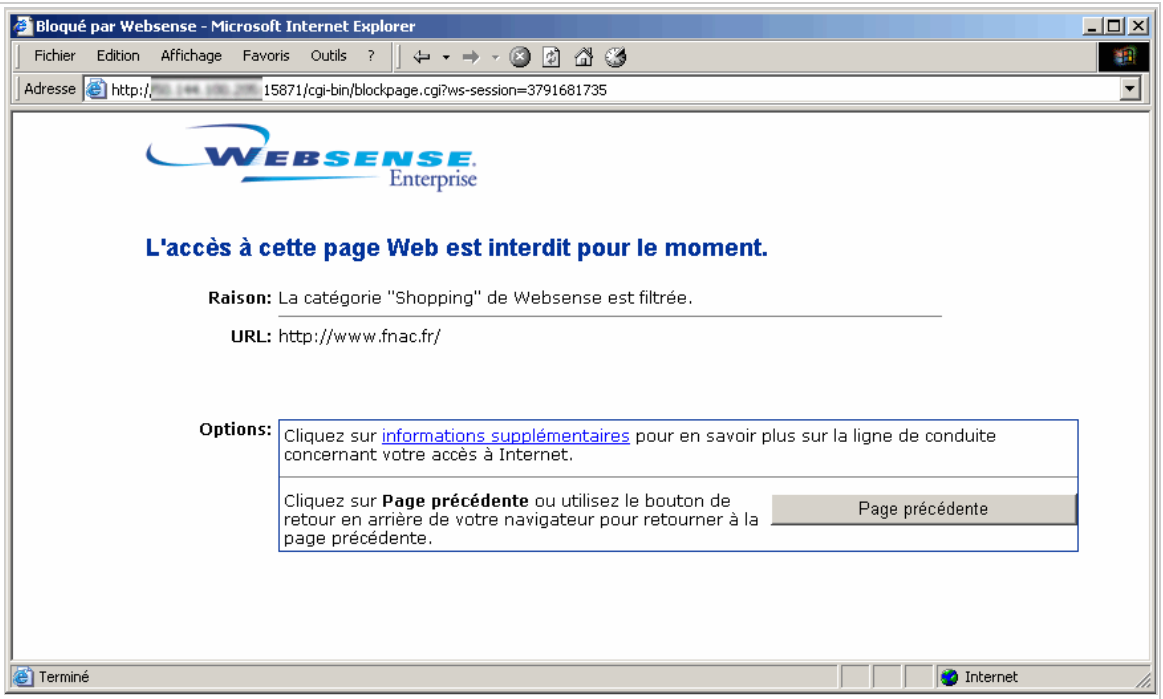
```
logdir /usr/local/squidGuard/log
dbhome /usr/local/squidGuard/db
src grownups { ip 10.0.0.0/24 user foo bar }
src kids { ip 10.0.1.0/24 }
dest porn { domainlist porn/domains urllist porn/urls }
acl {
  grownups { pass all }
  kids { pass !porn all }
  default {
     pass none
     redirect http://info.foo.bar/cgi/blocked?clientaddr=%a&clientname=%n&clientuser=%i&clientgroup=%s&targetgroup=%t&url=%u
  }
}
```

Le seul point délicat est la présence impérative d'une page de redirection accessible par Squid pour signaler à l'utilisateur l'interdiction d'accès. Si on souhaite éviter la configuration manuelle des différents logiciels mentionnés ici (Squid et SquidGuard) il est possible sous Unix, notamment pour le premier, de disposer d'un outil de configuration plus convivial au travers du système d'administration Webmin (http://www.webmin.com/). Un guide de configuration convivial et détaillé est disponible en français sur ce sujet (http://christian.caleca.free.fr/squid/).

Relais commerciaux et filtrage

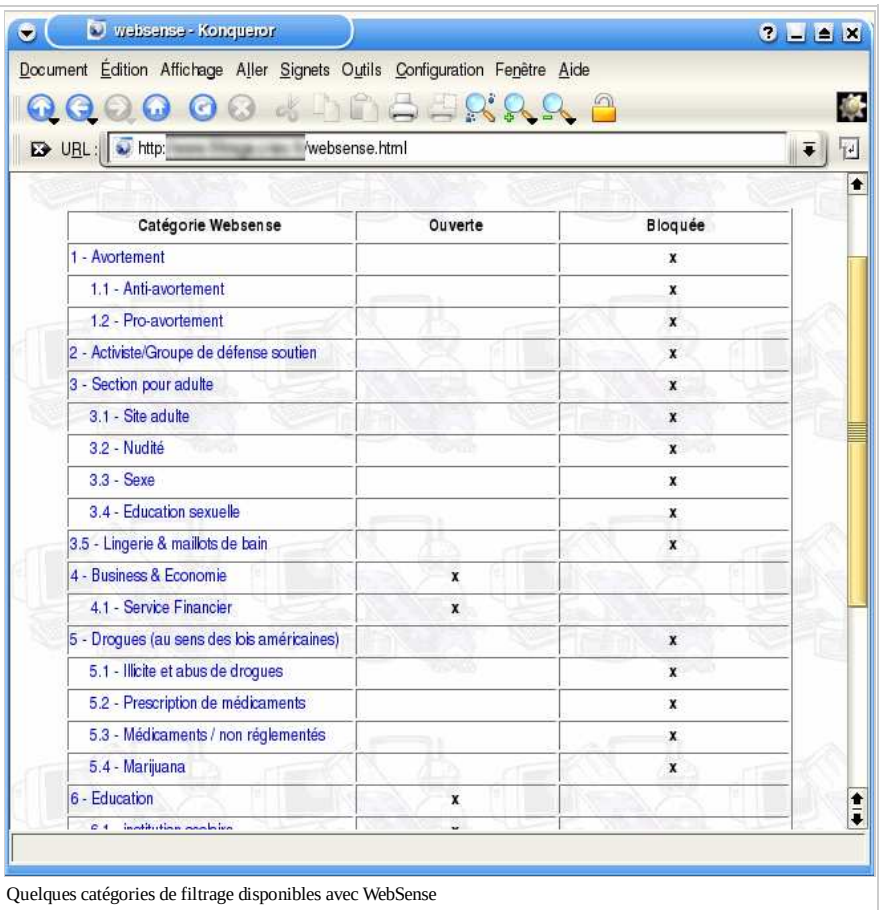
Les relais HTTP disponibles dans le domaine commercial fonctionnent généralement de la même manière que le couple Squid+SquidGuard que nous avons déjà détaillé. Ils s'appuient sur un logiciel tiers dédié à la réalisation du filtrage d'URL.

La figure suivante présente un exemple de filtrage réalisé par le logiciel WebSense (http://www.websense.com/) lors d'un accès HTTP au travers d'une infrastructure de proxy HTTP utilisant Microsoft ISA (http://www.microsoft.com/isaserver/).



Page d'interdiction du logiciel de filtrage WebSense

La figure suivante présente l'état de la configuration de filtrage mise en place dans ce cas particulier pour WebSense. Même avec une liste partielle, on note le nombre important de catégories de site Web disponibles par rapport à des listes noires librement disponibles. La fréquence des mises à jour est aussi probablement plus importante.



Catégorie Websense	Ouverte	Bloquée
1 - Avortement		x
1.1 - Anti-avortement		x
1.2 - Pro-avortement		x
2 - Activiste/Groupe de défense soutien		x
3 - Section pour adulte		x
3.1 - Site adulte		x
3.2 - Nudité		x
3.3 - Sexe		x
3.4 - Education sexuelle		x
3.5 - Lingerie & maillots de bain		x
4 - Business & Economie	x	
4.1 - Service Financier	x	
5 - Drogues (au sens des lois américaines)		x
5.1 - Illicite et abus de drogues		x
5.2 - Prescription de médicaments		x
5.3 - Médicaments / non réglementés		x
5.4 - Marijuana		x
6 - Education	x	

Quelques catégories de filtrage disponibles avec WebSense

Sécurité informatique/Signature et messagerie

OpenPGP et la confiance

- Un protocole : OpenPGP [RFC 2440].
- Deux principales implémentations : PGP et GnuPG.
- Le conteneur contient : un bi-clef, un ensemble de signatures et des informations « administratives ».
- Signer une clef :
  - cela signifie que vous avez pu vérifier directement l'identité du détenteur de la clef publique (par exemple à l'aide d'un hash de cette clef communiqué en personne et d'une carte d'identité) ;
  - cela ne signifie rien d'autre.
- Ce sont des systèmes utilisables pour signer ou chiffrer des fichiers et des messages (ou les deux simultanément).
- *Trust* : C'est paramètre permettant de limiter la transitivité (et indiquer ceux qui ne définissent pas « signer une clef » comme vous).

S/MIME

Sécurité informatique/DNS avec BIND 9

La figure ci-dessous présente le fonctionnement général des requêtes DNS dans une architecture comportant plusieurs serveurs DNS. Certains de ces serveurs (IP 192.168.69.1, 192.168.66.1 et 194.54.3.68) sont associés à la gestion des noms d'un domaine fictif `test.fr`. Un autre serveur DNS privé (IP 10.59.1.3) est associé à la gestion d'un domaine fictif privé (invisible depuis Internet) nommé `here.local`. On a également représenté deux serveurs DNS externes quelconques situés sur Internet.

Cette architecture correspond à une situation où les serveurs DNS internes de l'entreprise sont installés dans la zone protégée du réseau, à la fois pour le domaine visible depuis Internet (`test.fr`) et pour son domaine privé local (`here.local`). Plusieurs serveurs DNS (deux au moins) devant être disponibles depuis Internet pour gérer le domaine public `test.fr` (c'est une obligation pour se voir attribuer un nom de domaine), un deuxième serveur est positionné dans une DMZ (192.168.66.1), et un troisième (194.54.3.68) est certainement situé chez un prestataire extérieur offrant un service de secours réellement redondant.

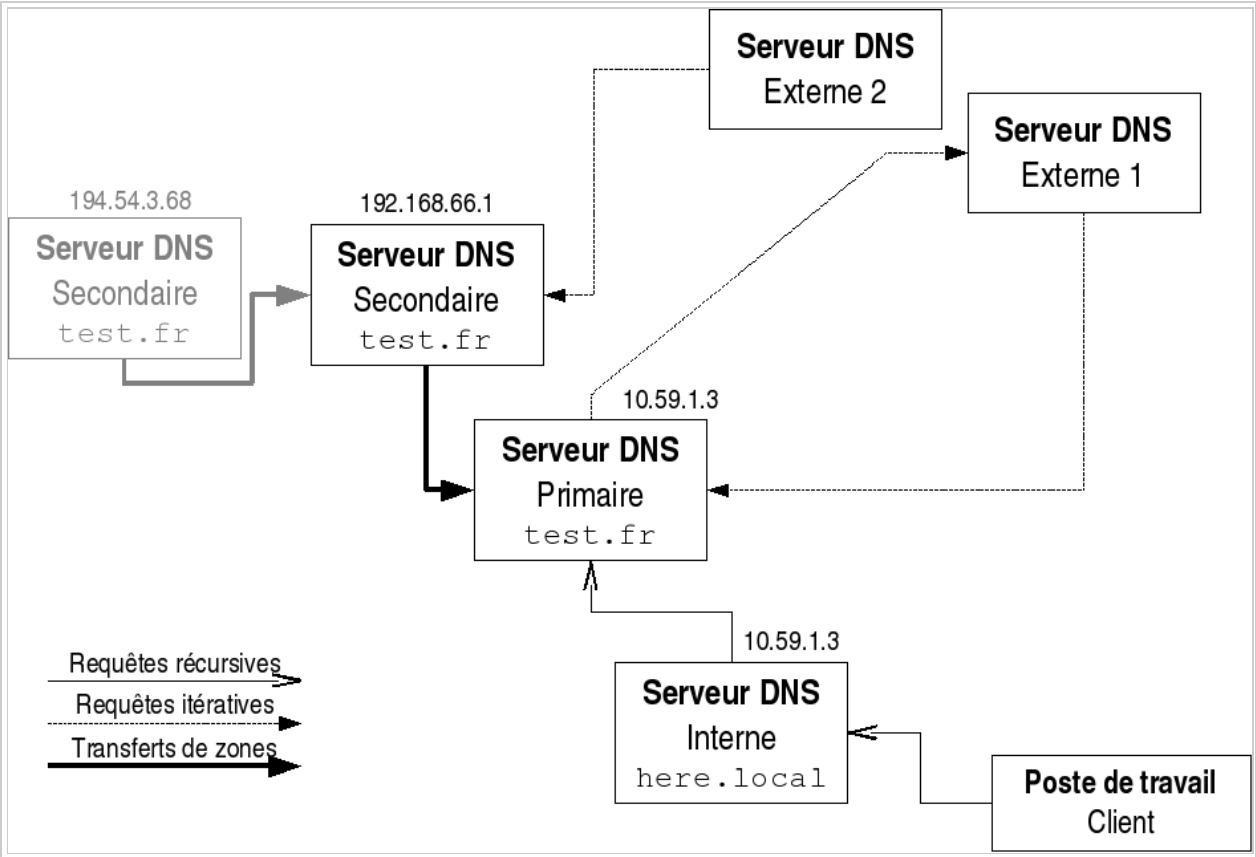


Schéma général des échanges DNS

Dans ce schéma, on distingue les différents types de requêtes DNS pouvant transiter sur le réseau :

- Les *requêtes récursives* correspondent aux demandes normalement effectuées par les clients à leur serveur DNS habituel, lequel prend alors en charge la totalité du protocole de résolution des noms. Il consulte son cache et contacte si besoin plusieurs autres serveurs DNS pour obtenir une réponse ou une erreur.
- Les *requêtes itératives* correspondent aux demandes de résolution effectuées généralement entre les serveurs DNS eux-mêmes. Les réponses aux requêtes itératives peuvent consister à rediriger le demandeur vers un autre serveur, charge au demandeur de poursuivre lui-même la résolution.
- Les requêtes de *transferts de zones* sont effectuées entre un serveur secondaire et le serveur primaire pour un même domaine pour permettre au serveur secondaire de posséder une copie à jour de la zone DNS qu'il gère.

On distingue également le double rôle que jouent généralement les serveurs DNS du point de vue des flux d'information traversant l'architecture. En effet, ils sont à la fois :

- des serveurs au sens propre du terme qui répondent aux demandes entrantes de clients externes (d'autres serveurs DNS) concernant le domaine géré (`test.fr` dans notre exemple) ;
- et des relais qui exécutent pour le compte de clients internes des demandes sortantes de résolution DNS et maintiennent aussi un cache pour les accélérer.

Une première approche de la sécurité des serveurs DNS consiste à clarifier l'architecture et bien paramétrer les serveurs de manière à ce qu'il répondent à ces différentes demandes correctement en fonction de l'origine du client.

Ainsi, il importe d'abord de limiter correctement les transferts de zones, qui diffusent la totalité des informations concernant `test.fr` en une seule fois. Cette limitation doit être faite sur le serveur primaire (IP 192.168.69.1) en y identifiant les serveurs secondaires autorisés à recevoir une copie de la zone :

```
zone "test.fr" {
  type master;
  file "/etc/bind/db.test.fr";
  allow-transfer {
    192.168.66.1;
    // or 194.56.3.68; (see below)
  };
};
```

Mais cette limitation peut également être effectuée sur le serveur secondaire (IP 192.168.66.1) en lui indiquant précisément le serveur primaire à utiliser pour la

zone `test.fr` dont il est secondaire :

```
zone "test.fr" {
  type slave;
  masters { 192.168.69.1; };
  file "/etc/bind/bak.db.test.fr";
  allow-transfer {
    194.56.3.68; // or "none"
  };
};
```

Ensuite, sur le serveur primaire comme sur le serveur secondaire, il est possible de contrôler les accès effectués aux zones gérées :

- les requêtes itératives proviennent d'autres serveurs (c'est à dire d'Internet) ;
- les requêtes récursives proviennent seulement des clients internes, c'est à dire des utilisateurs finaux (ou de leur mandataire).

En effet, dans l'exemple que nous suivons le serveur DNS interne public à l'adresse IP 192.168.69.1 reçoit en fait les requêtes des utilisateurs via le serveur DNS interne privé à l'adresse 10.59.1.3. Ce dernier est en fait normalement son seul client, à part pour quelques exceptions (autres serveurs, dépannages éventuels, etc.). Ce sont ces règles que nous appliquons ici :

```
// We allow only recursive queries from the internal nameserver, the 2nd, and self
acl "ns_rzo" { 192.168.66.1; 10.59.1.3; 127.0.0.1; };
// We also allow some admin. station to do queries here directly in case of problem
acl "admin" { 192.168.65.1; };
…
  allow-query { any; }; // or "slaves_ns"
  allow-recursion { "ns_rzo"; "admin"; };
```

Nous avons donc ici le cas d'un serveur DNS (IP 192.168.69.1) qui contient la référence du domaine public d'une entreprise (`test.fr`), qui gère pour le compte de tous les postes de travail la résolution des requêtes DNS sur Internet mais dont la configuration ne permet quasiment qu'un seul client. C'est en fait parfaitement conforme au cheminement souhaité des flux réseau dans l'architecture. Par ailleurs, le serveur DNS interne privé (IP 10.59.1.3) étant très proche du serveur DNS accédant à Internet, il n'est pas nécessaire d'utiliser ses fonctions de cache qui sont redondantes avec celles mises en œuvre au niveau suivant. Il est alors utile de faire fonctionner ce dernier serveur en mode relais pur :

```
options {
…
  // Allowed forwarders (only the DMZ nameservers)
  forwarders {
    192.168.69.1; 192.168.66.1;
  };
  // We *always* forward
  forward only;
…
};
```

L'intérêt de ce type d'architecture est de permettre une protection maximale du serveur DNS interne privé de l'entreprise qui n'accède absolument pas à Internet directement mais qui est cependant en mesure d'assurer l'ensemble des services DNS nécessaires aux postes de travail du réseau local, y compris la résolution de noms externes à l'entreprise. Ce serveur est aussi en mesure de résoudre les noms internes à l'entreprise (dans le domaine here.local) qu'il gère directement. Les postes de travail accèdent donc de manière transparente à tous les différents domaines.

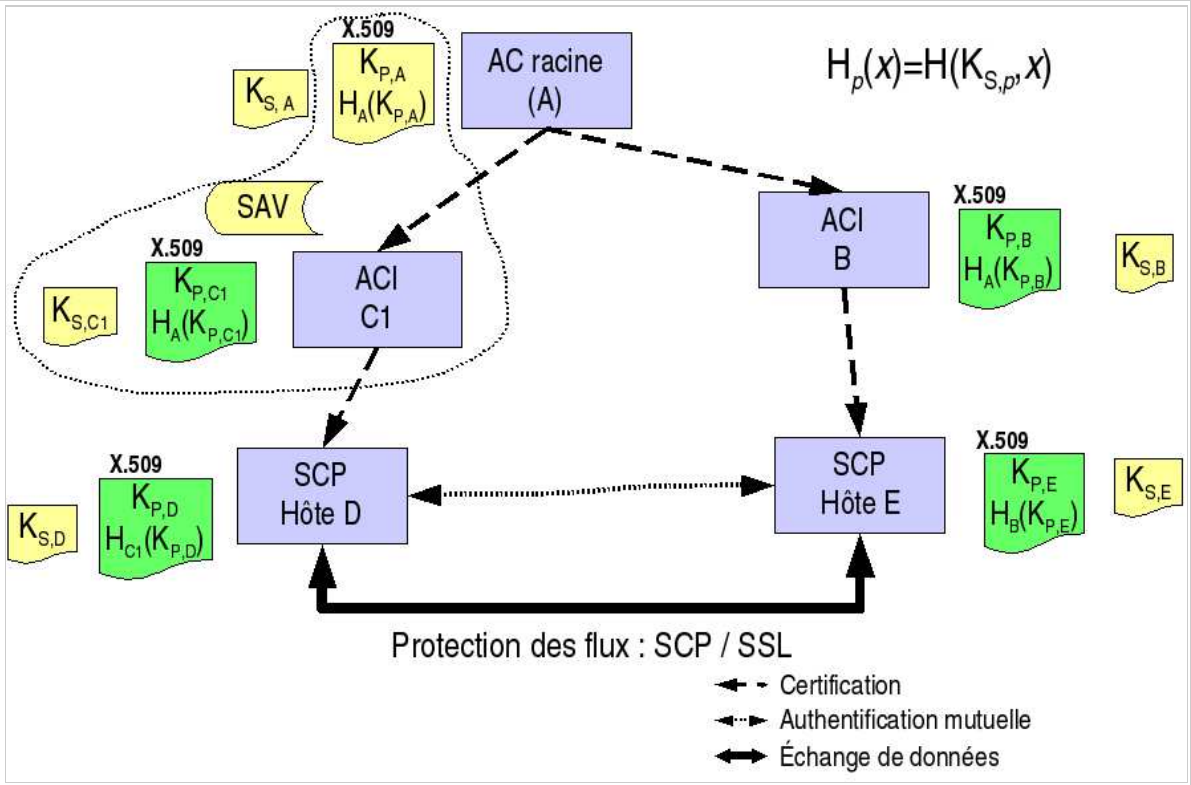
Sécurité informatique/PKI

À plusieurs reprises durant l'examen de certains logiciels, nous avons mentionné la possibilité d'utiliser des certificats X.509 pour vérifier l'identité des utilisateurs ou des services offerts sur Internet ou dans un réseau d'entreprise. Il est donc également nécessaire d'aborder en pratique le problème de la génération et de la gestion de ces certificats.

Ce sujet correspond au thème de la mise en place d'une infrastructure de gestion de clefs pour des algorithmes de cryptographie asymétrique, souvent désigné par l'acronyme PKI (pour *Public Key Infrastructure*). C'est un thème bien évidemment plus large que celui que nous souhaitons aborder dans ce document. C'est même un thème parfois très large dans la littérature où les ambitions pharaoniques occultent parfois les possibilités de réalisation concrètes encore relativement limitées (mais très utiles si on veut se donner la peine de les utiliser et de considérer leurs limites).

Outre les infrastructures disponibles dans certaines offres commerciales qui sont à envisager dans le cadre de projets de grande envergure, le principal moyen de générer des certificats X.509 (et donc de disposer d'une PKI) reste l'utilisation des utilitaires du projet OpenSSL (http://www.openssl.org). openssl est même à notre sens la première source d'information disponible pour traiter des certificats X.509 dans les contextes opérationnels concrets.

Une infrastructure de clefs utilisant les possibilités offertes par le format X.509 est présentée dans la figure suivante pour une application simple de transfert de fichiers via SCP avec authentification par certificats. L'intérêt du fonctionnement avec X.509 est de permettre à une autorité de certification racine (ici A) de déléguer les possibilités de création de certificats à d'autres entités dites autorités de certification intermédiaires (ici B et C). Par la suite, chacune de ces autorités intermédiaires est en mesure de générer des certificats utilisateurs permettant d'identifier respectivement les machines D et E pour faire fonctionner l'application désirée. Une infrastructure hiérarchique de ce type (en tout cas permettant la délégation de la création des certificats) est bien évidemment indispensable dans les cas de grandes organisations où un nombre important de certificats doivent être délivrés1. Par contre, à notre sens, le problème de la révocation des certificats X.509 (avant leur date d'expiration) est largement peu résolu pour l'instant dans les solutions PKI. En effet, il revient à utiliser et distribuer une liste centralisée de certificats révoqués qui vient contrecarrer les opportunités de déploiement à grande échelle promises par l'infrastructure hiérarchique de génération des certificats.



Exemple d'infrastructure X.509

Certains projets appuyés sur OpenSSL commencent à apparaître dans le domaine des logiciels libre pour aborder le problème de la mise en place d'une PKI dans son ensemble (avec tous les outils de gestion adaptés). Nous pouvons mentionner notamment un projet initié par une société française : IDX-PKI (http://idx-pki.idealx.org/) et un projet encore dans un état moins opérationnel : OpenCA (http://www.openca.org/).

Une liste intéressante (sans doute partielle) des autorités de certification reconnues ouvertement sur Internet est également disponible à l'adresse : http://www.pki-page.org/ qui montre que le sujet commence à trouver des débouchés concrets.

Sécurité informatique/Outils de recherche de vulnérabilité

Une dernière catégorie d'outils utiles à la gestion de la sécurité d'un système informatique est celle constituée par les outils de recherche de vulnérabilités, souvent appelés des outils d'audit du système.

Ces outils permettent d'effectuer des analyses du système d'exploitation, des logiciels ou de la configuration d'un système informatique, éventuellement à distance, et produisent des rapports identifiant les éventuels problèmes de sécurité rencontrés dans le périmètre soumis à leur recherche. Différents types d'outils de recherche existent dans ce domaine, depuis les précurseurs (notamment COPS sous Unix) jusqu'à des solutions ou des services commerciaux actuels. Ces offres externes peuvent d'ailleurs être complétées par des scripts spécifiques réalisés par les administrateurs du système eux-mêmes pour surveiller certains points précis (présence d'un virus particulier par exemple).

Nous nous intéresserons plus particulièrement dans cette section à deux outils de la catégorie des outils de recherche de vulnérabilités via le réseau, probablement la plus répandue). L'un d'entre eux, Nessus (http://www.nessus.org/) est diffusé dans le domaine du logiciel libre et a connu un succés grandissant. L'autre, Internet Scanner d'ISS est un des premiers outils de ce type a avoir été diffusé avec succés dans le domaine commercial.

Nessus

Nessus (http://www.nessus.org/) est l'outil de recherche de vulnérabilités le plus connu dans le domaine des logiciels libres. Ce logiciel est focalisé sur la recherche de vulnérabilités très variées sur les systèmes et dispose d'un mécanisme d'extension permettant d'ajouter assez facilement l'analyse de nouvelles vulnérabilités (via des plugins). Nessus effectue ses analyses à partir d'un serveur Unix situé sur le réseau qui constitue un point central pour l'administration du logiciel et le stockage des informations collectées. L'essentiel des tests de présence d'une vulnérabilité sont donc effectués à partir du réseau (bien que la nouvelle version 2.2 inclus également des fonctions d'analyse locale exécutées à distance au travers d'une session SSH) Des interfaces graphiques permettent d'accéder au serveur Nessus via le réseau (par des connexions SSL généralement authentifiées à l'aide d'un certificat) et de piloter l'exécution des sessions d'analyse (ou scan). L'interface graphique native fonctionne sous Unix/X11 mais des implémentations existent également pour MS/Windows (comme NessusWX ou NeWT 2.0).

À partir de l'interface de Nessus, on peut accéder successivement aux différents paramètres utilisés au cours du scan et exploiter les résultats obtenus. La page de démonstration du serveur Web du projet Nessus (http://www.nessus.org/demo/), illustre le déroulement d'une session d'analyse utilisant ce logiciel.

Tout d'abord l'écran initial de l'interface permet d'indiquer le serveur Nessus utilisé et le mot de passe permettant de s'authentifier auprès de ce serveur, soit directement, soit en déverrouillant la clef privée du certificat utilisé. (Bien entendu, une phase préalable de déclaration des utilisateurs et de mise en place des certificats doit être effectuée.) Une fois la connexion établie, on peut accéder aux autres paramètres.

Plusieurs écrans permettent ensuite de sélectionner les vulnérabilités à tester lors de la recherche et de fournir des paramètres spécifiques aux différents types de vulnérabilité recherchés (par exemple les comptes à utiliser pour certains tests). Les vulnérabilités testées par Nessus étant très nombreuses elles sont regroupées par catégories (CGI abuses, FTP, Windows, Backdoors, etc.) pour faciliter la sélection. Une particularité de Nessus est de réaliser un test de vulnérabilité le plus complet possible. Si ce n'est pas toujours systématique, c'est du moins l'esprit dans lequel sont développés les vérifications. L'exécution de certains tests peut donc réellement tenter d'exploiter une vulnérabilité, avec les effets secondaires que cela peut impliquer en terme de perturbation des machines (arrêt d'un service réseau, erreur du système d'exploitation). Ces tests risqués sont repérés explicitement et il est généralement recommandé de les désactiver pour un système informatique en exploitation.

Enfin, un autre onglet de l'interface permet de définir la plage d'adresses IP cible de la recherche de vulnérabilités. Associé à ces définitions réseau on trouve également les paramètres de déroulement du scan réseau précédent les tests individuels, ainsi que certaines options d'optimisation ou d'identification. Enfin, il est possible d'enregistrer la session d'analyse au niveau du serveur. Ceci permet notamment d'interrompre puis de reprendre une session particulière, mais également d'effectuer dans une certaines mesures des tests incrémentaux pour rechercher l'apparition de nouvelles vulnérabilités.

L'interface présente ensuite le déroulement d'un scan. L'exécution se déroule de manière concurrente sur plusieurs cibles simultanément. Pour chaque cible, une première phase de scan des ports réseau ouvert est exécutée, suivie par la recherche effective des vulnérabilités sur les services réseau identifiés. (L'exécution de tests locaux à un système n'est pas illustrée dans la démo car c'est une fonctionnalité encore assez récente à ce jour.)

Enfin, l'interface fourni un accès aux informations collectées par Nessus. Nessus permet aussi de générer un certain nombre de rapports dans divers format (notamment HTML) listant l'ensemble des vulnérabilités identifiées. Toutefois, pour un système informatique de taille usuelle comptant plusieurs dizaines de machines, les informations contenues dans un tel rapport peuvent être extrêmement nombreuses et l'exploitation des résultats nécessite un outil permettant de faire facilement des recherches. Certains patches pour Nessus permettent de rassembler les vulnérabilités identifiées dans une base de données (notamment par rapport au projet Prelude-IDS) mais l'exploitation de la masse d'information collectée par un tel outil n'est pas toujours facile à réaliser. C'est toutefois un problème commun à la plupart des outils de ce type dès qu'ils atteignent une certaine exhaustivité par rapport aux problèmes de sécurité référencés.

ISS Internet Scanner

Un des premiers outils commerciaux dans le domaine de la recherche de vulnérabilité via la réalisation de scan réseau a été *Internet Scanner* d'ISS (http://www.iss.net/). C'est toujours un outil de référence dans ce domaine, avec un nombre affiché de vulnérabilités prises en compte extrêmement important et des procédures automatiques de mise à jour des tests de vulnérabilité. Celles-ci sont similaires à celles utilisées par la sonde de détection d'intrusion du même éditeur (*RealSecure*). Les deux produits bénéficient d'ailleurs de la mutualisation des efforts de prise en compte des vulnérabilités au sein du même éditeur (soit pour la détection de l'attaque, soit pour le test de présence).